

University of Economics, Prague

Faculty of Informatics and Statistics



**WEB APPLICATION TO SUPPORT AGILE
PROJECT MANAGEMENT PRACTICES IN
THE LARGE ENTERPRISE PROJECT WITH
A VIRTUAL TEAM**

MASTER THESIS

Study programme: Applied Informatics

Field of study: Information Technologies

Author: Bc. Victoria Panchenko

Supervisor: Ing. et Ing. Soňa Karkošková, Ph.D.

Prague, December 2019

Declaration

I hereby declare that I am the sole author of the thesis entitled “Web Application to Support Agile Project Management Practices in the Large Enterprise Project with a Virtual Team”. I duly marked out all quotations. The used literature and sources are stated in the attached list of references.

In Prague on

Signature

Student's name

Acknowledgement

I hereby wish to express my appreciation and gratitude to the supervisor of my thesis Ing. et Ing. Soňa Karkošková, Ph.D., same as to all lecturers in the University of Economics in Prague, which I met during my studies. I want to express appreciation to my family and close friends, who had supported me throughout my studies.

Abstract

The boom of agile methodologies contributed to increase of the number of large enterprises that are adopting agile practices. On the other hand, the last decades have witnessed a steady trend toward globalization and global software development has become a common business reality. Global virtual teams do not correspond with agile approach of collocated teams, and the implementation of agile methods in distributed teams still presents challenges. This Master thesis is dedicated to the design and development of web application for the case project with a virtual agile team. The application is created to improve current agile practices on the large enterprise project with a geographically dispersed team.

Keywords

three-tier architecture, MVC, agile practices, virtual teams, real-time application

JEL Classification

three-tier architecture, MVC, agile practices, virtual teams, real-time application

Content

Introduction	12
1 Current state of problem area	16
1.1 Scientific articles and research papers	16
1.2 Bachelor and Master theses, academic publications	19
1.3 Specialized literature	21
2 Research plan and Methodology	23
3 Agile project management	26
3.1 Comparison of traditional and agile methodologies	26
3.2 State of Agile report	28
3.3 Scrum framework	29
3.3.1 Roles	30
3.3.2 Artifacts	30
3.3.3 Ceremonies	32
3.4 Summary	34
4 Current agile practices on the case project	35
4.1 Current client's solution	35
4.2 Competing ready-made solutions	37
4.2.1 RealtimeBoard	37
4.2.2 IdeaBoardz	38
4.2.3 Pointing Poker	39
4.2.4 Fire Poker	40
4.3 Summary	41
5 Analysis and requirements' establishment	42
5.1 Initial discussion with the client	43
5.1.1 General requirements on the application	43
5.1.2 Retrospective	43
5.1.3 Brainstorming	44
5.1.4 Poker Planning	44
5.2 Discussion with the project managers and Scrum Masters	45
5.2.1 Asked questions and frequent answers	45
5.2.2 Concluding propositions from the discussion	46
5.3 Usage of the web application	46
5.4 List of the final requirements	46

5.4.1	General functional requirements	47
5.4.2	Functional requirements on the Retrospective feature.....	47
5.4.3	Functional requirements on the Poker Planning feature.....	48
5.4.4	Functional requirements on the Brainstorming feature	48
5.4.5	Non-Functional Requirements.....	49
5.5	Summary.....	49
6	Design of the application prototype	50
6.1	Use Cases	50
6.1.1	Actors.....	51
6.1.2	General Use Cases	51
6.1.3	Retrospective Use Cases	53
6.1.4	Poker Planning Use Cases	55
6.1.5	Brainstorming Use Cases	58
6.1.6	Requirements traceability matrix.....	60
6.2	Entity relationship diagram.....	63
6.3	Application's architecture	65
6.4	User Interface design.....	66
6.4.1	Retrospective event GUI.....	68
6.4.2	Poker Planning – creating User Story phase.....	70
6.4.3	Poker Planning – voting phase.....	71
6.4.4	Poker Planning – discussion phase.....	73
6.4.5	Brainstorming event GUI	74
6.5	Client's remarks and adjustments	76
6.6	Summary.....	77
7	Implementation	78
7.1	Architecture	78
7.2	Used libraries, frameworks and technologies.....	84
7.2.1	ASP.NET MVC.....	84
7.2.2	ASP.NET Identity and OWIN.....	84
7.2.3	SignalR.....	85
7.2.4	Entity Framework.....	86
7.2.5	LINQ	86
7.2.6	EPPlus.....	88
7.2.7	Frontend technologies.....	88
7.3	Hosting.....	89

7.4	Summary.....	89
8	Evaluation and future improvements	90
8.1	Acceptance testing results	90
8.2	Main implications	93
8.3	Limitations.....	94
8.4	Suggestions for future improvements	96
8.5	Summary.....	97
	Conclusions	99
	List of references	101
	Annexes	I
	Annex A: UML Use Case Diagrams	I
	Annex B: Physical Entity Relationship Diagram, created in EA	IV

List of Figures

Figure 1 Method Framework for Design Science Research (Johannesson, P. et al., 2014)	23
Figure 2 Scope, resources and time in traditional versus agile approach (Ca.gov, 2017)...	27
Figure 3 Agile Methodologies used by respondents (CollabNet VersionOne, 2019)	28
Figure 4 Scrum Values (Scrum.org, 2019)	29
Figure 5 Product Backlog (Visual-paradigm.com, 2018)	31
Figure 6 Sprint Backlog (Visual-paradigm.com, 2018)	32
Figure 7 Scrum framework at a glance (Neon Rain Interactive, 2015)	34
Figure 8 Retrospective feature of RealtimeBoard	37
Figure 9 Retrospective feature of IdeaBoardz	38
Figure 10 An estimating session from the Pointing Poker application	39
Figure 11 Fire Poker application	40
Figure 12 Requirements establishment process	42
Figure 13 Traceability matrix for the general requirements	61
Figure 14 Traceability matrix for Retrospective requirements	62
Figure 15 Traceability matrix for Poker Planning requirements	62
Figure 16 Traceability matrix for Brainstorming requirements	63
Figure 17 Process of creating ERD in Enterprise Architect software	64
Figure 18 A three-tier architecture (Kaushik, 2016)	66
Figure 19 Home page of "CPS Agile" application	67
Figure 20 Page to create an event	68
Figure 21 "Preparing points" phase from a regular user's perspective	69
Figure 22 "Preparing points" phase from a moderator's perspective	70
Figure 23 Poker Planning - User Story creation page, moderator's perspective	71
Figure 24 Poker Planning - voting phase, moderator's perspective	72
Figure 25 Poker Planning - voting phase, regular user's perspective	72
Figure 26 Poker Planning - discussion phase, moderator's perspective	73
Figure 27 Poker Planning - discussion phase, regular user's perspective	74
Figure 28 Brainstorming event from a moderator's perspective	75
Figure 29 Brainstorming event from a regular user's perspective	75
Figure 30 Final view of presentation phase after client's remarks	76
Figure 31 Interaction within MVC pattern	79
Figure 32 DIP, IoC, DI and IoC Container (Tutorialsteacher.com, 2019)	80
Figure 33 SignalR Hub communication (C-sharpcorner.com, 2019)	81
Figure 34 Structure of CpsBoostAgile project, business logic layer	83
Figure 35 DbContext responsibilities in EF (Entityframeworktutorial.net, 2019)	84
Figure 36 SignalR client-server communication (PubNub, 2019)	85
Figure 37 EF in application's architecture (Entityframeworktutorial.net, 2013)	86
Figure 38 Retrieving data with LINQ (GeeksforGeeks, 2019)	87
Figure 39 Modification in the work with the items after the first testing	90
Figure 40 Modification in User Story creation	91
Figure 41 Modified "Coffee break" functionality	92
Figure 42 New requirement about Available votes, changes in GUI	93

List of abbreviations

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
ASD	Agile Software Development
BFR	Brainstorming Functional Requirement
BGP	Border Gateway Protocol
BLL	Business Logic Layer
BUC	Brainstorming Use Case
CSS	Cascading Style Sheets
CSV	Comma Separated Values
DAO	Data Access Object
DAL	Data Access Layer
DBMS	Database Management System
DI	Dependency Injection
DNS	Domain Name System
DOM	Document Object Model
DSDM	Dynamic Systems Development Method
EA	Enterprise Architect
EF	Entity Framework
ERD	Entity Relationship Diagram
FDD	Feature Driven Development
FTFP	Fixed Time Fixed Price
GDPR	General Data Protection Regulation
GFR	General Functional Requirement
GSD	Global Software Development
GUC	General Use Case
GUI	Graphical User Interface

GUID	Globally Unique Identifier
HTML	Hypertext Markup Language
ICT	Information and Communications Technology
IDE	Integrated Development Environment
IIS	Internet Information Services
IMRaD	Introduction – Method – Results – and – Discussion
IoC	Inversion of Control
ISP	Interface Segregation Principle
IT	Information Technology
JS	JavaScript
LINQ	Language Integrated Query
MS	Microsoft
MVC	Model-View-Controller
ORM	Object-Relational Mapping
OWIN	Open Web Interface for .NET
RDBMS	Relational Database Management System
PDF	Portable Document Format
RFR	Retrospective Functional Requirement
PPBFR	Poker Planning Functional Requirement
PPUC	Poker Planning Use Case
QA	Quality Assurance
RPC	Remote Procedure Call
RUC	Retrospective Use Case
RUP	Rational Unified Process
SLR	Systematic Literature Review
SQL	Structured Query Language
SSL	Secure Sockets Layer

TLS	Transport Layer Security
UI	User Interface
UML	Unified Modeling Language
XML	Extensible Markup Language
XP	Extreme Programming

Introduction

Nowadays the number of large enterprises that are adopting agile methodologies is increasing. However, organizations are looking for cheaper and faster ways to develop software and for this reason are moving to global software development. Global virtual teams are an opposite of face-to-face collaboration and co-location agile principles. Therefore, the adoption of agile practices in geographically dispersed teams is still a challenging process. This Master thesis presents the case project with virtual agile teams. The teams develop and maintain a large enterprise project, which was recently switched from traditional to agile project management. The main goal of this paperwork is to develop a web application, which improves current agile project management practices on the case project. The process of obtaining requirements, designing the application and developing a prototype is thoroughly described in the thesis. The last chapter is devoted to prototype's validation and the results of changed agile processes on the case project.

Background of topic

Unicorn a.s. is a large European company, which provides information systems and solutions in the area of Information and Communication Technologies. The company was established in Czech Republic in 1990. Over 1500 employees and over 300 European industry leading companies as customers make Unicorn the leading and largest Czech provider of Information and Communication Technology services (Unicorn, 2019). The main fields of business, where Unicorn a.s. delivers and distributes IT solutions, are banking, insurance, energy and utilities, government and public sector. Due to company's long-standing operations on the IT market and long-term collaboration with the clients, the company has gained valuable experience, large knowledge base and knowledge artifacts. Therefore, the firm has the unique opportunity to closely monitor and sometimes to participate in changes, which the clients have experienced over the time. In particular, the way of work and collaboration is evolving, as its basic traits and attributes are modifying, organizational structure and the corporate culture in companies are changing drastically. This was a case of a banking company, to which Unicorn a.s. has been delivering software solutions since 1998. The bank decided to follow trends in the IT sphere and to transit the financial institute, its organizational structure, company's culture from traditional to agile processes. This client's decision had determined consequences on the collaboration with Unicorn a.s. and especially on the case project. Before shifting to agile processes, the project ran on Fixed Time Fixed Price (FTFP) base and was managed using traditional waterfall methodology. This kind of pricing model assumes that the total budget on the project is agreed before development starts and it remains unchanged. This means a fixed amount of work, fixed budget and fixed deadline. The risks for underestimating the scope of work and not meeting a deadline is held by the contractor. A fixed price contract seems convenient for customers, as they know how much time and money the implementation of all requirements will cost. But it as well has its cons, for instance, higher development rates, because the contractor, working on FTFP base, puts additional costs to the budget to ensure

himself from contingencies. Another drawback is the inability to make additions or modifications after the contract is signed, or a complicity to make such modifications. The process of making adjustments to a scope of work requires an official change request and time demanding approval process, which results in higher costs for the business. All these disadvantages of FTFP contract and new trends in IT world reflected in a bank's desire for changes and a transition to Time and Material pricing model with the implementation of the agile project management practices. The financial institution, as the most companies, follows two goals. Firstly, to keep or reduce its internal costs, including the costs of creating new services and products, and secondly, to have the ability to take new services and products to market as quickly as possible. With the transition to agile processes and practices the top management of the bank expects acceleration of the development of existing products and the delivery of new products, while at the same time the transparency of the expenses will increase and the management will have a better understanding and better control over the costs.

Problem area

In the area of software development changes happen quite often and fresh advanced artifacts and instruments are quickly evolving. Therefore, when the original traditional methodologies have ceased to be suitable for use in some projects, they have started to be replaced by new methodologies that were initially described as light and later agile. The case project of my paperwork is an instance of this particular situation, when top management of a bank decided that a large enterprise project, which has been running since year 2001, needs to be moved forward and to be switched from traditional waterfall model to agile project management methodology, in particular Scrum framework. Nevertheless, the project's initial state should be always considered first, and it is necessary to evaluate if a project is in a suitable state to transit from traditional model to a light methodology. Unfortunately, when a change was introduced and implemented by the top management of the bank, the case project was not perfectly prepared for the shift due to its team size and geological dispersion. Caused by globalization, cost optimizing and rapidly developing technical resources, distributed teams are becoming a growing trend in the software development industry. But the benefits of virtual teams are coming with some trade-offs. For many remote teams it is difficult to adopt the agile practice of face-to-face interactions. A distributed team, who is working on the case project, was not ready for the new modified project management practices and therefore a transition was met as unwanted and unwilling accepted permutation. The present-day opinion of the management and team members is that overall improvement of currently used agile processes can boost the transition, bring comfort and convenience and dissolve employees' resistance to change. That brings us to the main purpose of my paper – to develop a web application, which supports agile project management practices in the large enterprise project with a virtual team.

Purpose of this paper

The main objective of my Master thesis is to develop a prototype of a web application, which contributes to agile project management practices and improves agile processes in a team working remotely. The prototype is developed for the case project and its team, who is engaged in a bank environment. The main purpose of this paper is achieved by meeting small particular goals such as analysis of a current state of the agile principles on the project, requirements' establishment from users and involved parties, detailed design of the prototype and the development itself. The prototype of the web application is then tested on the large enterprise project, which is run by a virtual team. One of the particular objectives is to validate the implemented prototype, evaluate the final results and provide suggestions for next future improvements. To reach the main goal of the paper I am using Design Science methodology. The methodology and its application are explicitly described in the second chapter "Research plan and Methodology" later in my paper.

Case project

Herein I would like to point out the main characteristics of the case project. Due to the signed contract, which reflects the General Data Protection Regulation (GDPR), I cannot name the financial institution and exact name of the project, which Unicorn a.s. delivers the team and the IT solution to. Therefore, here and later in my paper I am referring the case project and its team members as the client, the contracting authority or the customer.

As it was described in the Problem area part, the financial institution transited the existed project from traditional waterfall approach to agile project management, particularly Scrum framework, at the end of October 2018. The project has been running since 2001 and the average allocation of a team member on the project is 5 – 7 years. Even that it is almost one year gone, since the change was introduced, the coworkers are still adapting to the transition and changed processes, as they were used to traditional approach for many years. The scope of the project is large, and the project can be for certainty defined as an enterprise project. The final IT solution of the case project is used in each branch and subsidiary of the bank and is primary meant for the employees of the bank. It means that a number of the final users is around two thousand people. The solution has a web client and can be described as an administration software, which manages and ensures the processing of the wide range of financial products such as loans, mortgages, credit cards, as well as current and savings accounts and far more. The solution is tightly integrated in the whole IT infrastructure of the bank and is highly contingent on another internal IT systems, as it is connected through bank's middleware. Due to the project's size there are four teams in total, which consist of approximately 65 external workers and 35 internal employees of the financial institution. In fact, each team is responsible for its own part of the project, the teams work independently and under normal circumstances do not cross in their scope of work. The team of the case project, I have been working with, is a virtual team with 14 members in it. A part of the team is allocated in Pilsen, another city in the Czech Republic. The positive side is that the team is quite stable and, as I mentioned before,

the average allocation of each team member is from five till seven years, so that colleagues know each other for a long time and there is no communication problem inside the team. But it is still hard to practice agile management methodology in a distributed team, especially such activities as the Retrospective and Poker Planning, where the presence of team members is essential. Therefore, the need for a digitalization of Scrum practices have raised and the idea to create a web application, which supports such practices, was met by the team with a positive stand.

Expected contribution

By achieving the main purpose of this paper and developing the web application the large improvement of the processes, which support agile approach, is expected on the case project. The contribution to agile processes' advancement is expected to be proved on the real enterprise project with a virtual team. Processes enhancement is expected to reflect in team's overall mood and attitude to not-so-long-time-ago established agile practices on the case project, which was before managed using the waterfall model.

Structure

I structured my Master thesis in accordance with IMRaD (Introduction, Methods, Results, and Discussion) norm. IMRaD is a common organizational structure for the scientific articles. In the Introduction part I am explaining the motivation behind the chosen topic and stating the main goal of my paper. I am also conducting a summary of existing research on the subject using Systematic Literature Review (SLR) methodology. I am providing introduction to the field and describing the current situation on the case project. In the Methods part of my paper I am illustrating and commenting on what I did in order to achieve the preset goal. Next the results of the conducted actions are reported. The final part is a Discussion on the results, where I am presenting the practical implications, summing up the main findings and limitations, and drawing the future perspectives.

Assumptions and preconditions of the paper

This Master thesis does not aim to describe and compare individual project management methodologies or to cover all agile software development principles and practices. Furthermore, the objective of the thesis is not to judge and evaluate if the case project was suitable for the transition to agile methodology and if the shift was made properly and correct from the customer's side, as it cannot affect the current situation on the case project. The paper deals only with the implementation of the software prototype, which aims to facilitate the current agile practices inside the distributed team, which works on the large enterprise project.

1 Current state of problem area

This chapter presents the results of the research, which was made by using Systematic Literature Review methodology. A focus of the review was put on a precise topic – Tools, practices and techniques for agile project management in large projects or projects with virtual teams. As for the methods of data collection I searched several specific databases using three, four search queries. The main utilized database sources for my research were EBSCO, ProQuest Central, ACM Digital Library and IEEEExplore. For review of the related Bachelor and Master theses I used online system Theses.cz. Theses.cz is a national registry of theses and a repository for plagiarism search, in other words it is a database of academic research works and at the same time it serves as a system to detect plagiarism. The Theses.cz is developed and operated by the Faculty of Science, Masaryk University in Brno. Specialized professional literature was searched with the help of Google search engine and electronic information resources of Library of the University of Economics in Prague. The results of the research were limited by the earliest year 2001, as the Agile Manifesto was originally published in February 2001 and since then new light, agile project management methodologies have started to gain in popularity. The systematic review was conducted only for literature in Czech and English languages. As a method of data presentation, the table of included papers was chosen, thereby the result of the research can be found further in the corresponding tables of particular sections.

1.1 Scientific articles and research papers

To search related scientific articles and research papers in ProQuest Central, ACM Digital Library and EBSCO, such queries as the “*agile practices in global software development*”, “*agile tools in virtual teams*”, “*agile in large scale*” and “*enterprise agile projects*” were used in different combination. The filtered results with a description of the most relevant to my paper articles are structured and presented below in Table 1 Review results: Scientific articles and papers.

Table 1 Review results: Scientific articles and papers.

Name of the publication	Author	Year	Description of the publication
Virtual by Design: How a Work Environment can Support Agile Distributed Software Development.	Lous, P., Tell, P., Michelsen, C.B., Dittrich, Y., Kuhrmann, M. and Ebdrup, A.	2018	The article presents a case study, how a company manages the challenges, which are related to the combination of the agile processes and the distributed development. The paper states, that in a virtual work environment a significant role belongs to continuous improvement of the processes and careful selection of the technologies and tools, which are used in agile practices.

Project Management Tools for Agile Teams.	Mihalache, A.	2017	The paper compares various existing tools for agile project management. It also specifies what criteria a comparison was made on and why these criteria are important. The article explores how agile practices help reduce common problems of global software development (GSD).
Agile Practices Reduce Distance in Global Software Development.	Holmström, H., Fitzgerald, B., Ågerfalk, P.J. and Conchúir, E.Ó.	2006	The authors studied two companies and conducted research, using such methods as workshops, seminars and interviews. Based on these two case studies, agile methodologies, specifically Scrum and Extreme Programming (XP), are found to be useful for reducing three dominating obstacles that have been associated with GSD – communication, coordination, and control problems.
Inter-Team Coordination Mechanisms in Large-Scale Agile.	Nyrud, H. and Stray, V.	2017	This research paper explores coordination between agile teams in large-scale software development. The authors investigated inter-team coordination mechanisms and found eleven mechanisms that they mapped into the five categories. The research was made by interviewing project members and observing 26 meetings in three different teams working on a larger digitization project.
Scrum Master Activities: Process Tailoring in Large Enterprise Projects.	Bass, J.M.	2014	This paper discovers the problems of agile development in large-scale enterprise projects and investigates the role of Scrum Master in addressing such issues. The study was made in eight international companies, where interviews with 46 practitioners were conducted.
Effort Estimation in Large-Scale Software Development: An Industrial Case Study.	Usman, M., Britto, R., Damm, L.-O. and Börstler, J.	2018	This scientific article explores effort estimation in large-scale distributed agile projects. The authors also analyze the accuracy of the effort estimation processes, identifying the factors that have impact on the planning and estimation accuracy.

In the article “Virtual by Design” (Lous et al., 2018) the authors examine the challenges of a virtual work environment combined with agile practices. Among the main challenges were determined the meetings at only one site, lacking team cohesion and shortage of transparency between participants. The article states that having meetings at only one site isolates the rest of the team members and damages the feeling of being united as a team.

Therefore, the distributed team members do not necessarily perceive themselves as the part of the team, which might result in the different views of a common goal. A shortage of transparency can cause poor sharing of knowledge and intensifies the lack of team cohesion. The authors come to conclusion, that the carefully selected practice-tool combinations increase effectiveness and facilitate each team member to fully participate in processes regardless of the different physical location. This article is relevant to my Master thesis, as the expected contribution of my paperwork is an improvement of the agile processes in virtual work environment due to custom-developed tool.

The next article “Project Management Tools for Agile Teams” (Mihalache, 2017) evaluates the existing agile tools, based on the specific criteria. The article claims that agile project management tools should be different from the traditional tools due to dynamic and adaptive nature of the agile practices. For the comparison of tools, the varied criteria were used. The criteria were selected based on the maximization of the business value. One of the mentioned criteria is the ability to assess the project and to evaluate performance constantly. The Retrospective agile practice fits that idea perfectly. Further important criteria were progress reports, which are easy-to-understand for stakeholders, and communication on updates inside a team. At the end of the article, the author declares, that organizations and teams should reveal, analyze and deal with the weaknesses and issues emerged from product’s complexity and growing sizes of teams first, before rushing into selection of the agile tools.

The research paper “Agile Practices Reduce Distance in Global Software Development” (Holmström et al., 2006) investigates and shows on the instance of two companies, how Scrum and Extreme Programming practices are beneficial and constructive for global software development. Geographical and sociocultural distances impose several communication, coordination, and control challenges within GSD teams. The authors examine and review how introducing agile practices had solved or reduced these issues. Agile practices helped the case companies to increase “teamness”, increase mutual understanding and collaboration within and between teams. The practices also facilitated collective ownership of code and extended visualization of shared project activities. The case companies did not use a complete set of agile methods, they rather discovered that mixing of different agile methods helped them organize and control the complex tasks of GSD in the most efficient way.

Nyrud, H. and Stray, V. in their research paper “Inter-Team Coordination Mechanisms in Large-Scale Agile” (Nyrud and Stray, 2017) examine and define eleven coordination mechanisms, which they map into the five categories: Impersonal mode, Personal mode (Vertical/Horizontal) and Group mode (Scheduled/Unscheduled). Coordination mechanisms such as Jira¹, Rules for Quality assurance (QA), Open work area are referred to Impersonal mode. Personal mode represented by ad hoc conversations. To the Scheduled

¹ Jira is a proprietary software product developed by Atlassian. Jira is used for issue tracking and project management, including agile techniques.

Group mode belong such agile practices as Stand-Up meeting, Retrospective, Demo, Sprint planning and Backlog grooming.

The paper “Scrum Master Activities: Process Tailoring in Large Enterprise Projects” (Bass, 2014) explores the role of the Scrum Master – a facilitator for Development Team, who is promoting adherence to agile practices and removing hindrances for team members. But in large enterprise projects Scrum Masters work together in scrum master teams, which are often geographically distributed. Scrum master teams span ten activities. Scrum of scrums facilitator coordinates work with the other scrum masters. The sprint planner is responsible for a user story picking in each sprint. The integration coordinator ensures the integration of code developed by individual teams. The learning coordinator oversees possibilities of continuing professional growth for each team member. The heat beat monitor is accountable for the coordination, that teams are productive and having a work progress. The process anchor promotes adherence to agile practices. The task estimator supports sprint planning and coordinates a task assignment to the teams. The impediment remover ensures that team members do not have any hindrances for work progress. The spike process coordinator evaluates and manages any unexpectedly complicated user stories that arise during a sprint. The test coordinator ensures that the final product meets quality requirements. Understanding the nature of these activities can help practitioners to adapt the agile practices to a large-scale enterprise project.

The main findings of the case study “Effort estimation in large-scale software development” (Usman et al., 2018) are that a two-stage estimation process, with re-estimation at the analysis stage, improves the accuracy of the estimations. The study shows that teams incline to underestimate rather than overrate required efforts. The factors, which have impact on the estimation, are a maturity of the team, a multi-site development, size and priority of an estimated task. The study says that less mature teams, requirements with a larger scope of work and requirements developed in geographically distributed teams are more likely to be underestimated and incur effort overruns. The paper also highlights an importance of involving all stakeholders in the estimation process. The authors come to conclusion, that a two-stage effort estimation process can improve the accuracy and address the challenges in large-scale agile software development. To improve the accuracy of estimations, one needs to consider a team maturity, a team distribution as well as requirements size and priorities.

1.2 Bachelor and Master theses, academic publications

To search related Bachelor and Master theses, I used Theses.cz web portal, which serves as a national register portal for academic works. Keywords “agile tools”, “tools for virtual agile team”, “practices in large scale agile” were used to conduct the research. The results, which were filtered on criteria of relevance to my paperwork, are presented in Table 2.

Table 2 Review results: Bachelor and Master theses.

Name of the publication	Description of the publication
Scrum process in a distributed team (Nekola, 2012).	The thesis describes Scrum methodology and compares it to traditional and other agile methodologies. The author also develops a software, which supports management of team backlog and is suitable for virtual teams. The software was implemented using Play 2.0 framework and the Scala programming language.
Impact analysis of the implementation of Scrum methodology in the banking company (Grombířová, 2019).	This Master thesis analyzes the impacts of the Scrum agile methodology implementation in the financial institution. The author identifies the positive and negative implications of the agile transformation. The thesis discovers current problems in the case study company and propose recommendations to mitigate the negative aspects of the transformation.
Agile development scalling (Hubený, 2019).	The author in his Master thesis observes a company, which has over 200 employees and which recently adopted the agile practices. The author explores the available frameworks for scaling agile and proposes recommendations for the enterprises, which wants to transit to agile development.
Software tools for supporting virtual project collaboration (Kiesel, 2012)	The thesis compares 32 existing software applications for project management of virtual teams. The comparison describes tools, services, licenses and prices. The author analysis implementation and usage of the software on a case project.
Analysis of agile software development methodology usage in work teams (Horváthová, 2019)	The author studies the advantages of using Scrum agile methodology on the case company - Kiwi.com. The thesis demonstrates an application developed on demand, which helps the developers to obtain and process the reflections on task estimations.

Nekola (2012) in his thesis “Scrum process in a distributed team” describes the Scum methodology and develops a software tool, which is used for backlog management. The developed software is suitable for using in distributed teams and is implemented using Play 2.0 framework and the Scala programming language.

Grombířová (2019) in her thesis “Impact analysis of the implementation of Scrum methodology in the banking company” conducts an extensive research of transformation to agile practices on case company – a Czech financial institution. The author reveals the issues, which occurred with the transformation, and proposes the solution to these problems. As the author states in the Introduction section of her work, the only limitation of the thesis is the data anonymization due to sensitive data, which could harm

the organization in case of its publication. Therefore, based on the agreement with the case company, any names, project name and organization name are omitted.

Hubený (2018) in the thesis “Agile development scalling” analyzes existing scaling frameworks, develops recommendations and provides useful information for successful transition for companies, which want to switch to agile development. The author studies the five most widely used scaling frameworks and observes a case company, which has successfully transited to agile approach. Based on his observation and research, the author develops recommendation for the enterprises that are speculating about the adoption of agile methodologies. Similar academic works are presented by Krotkov (2019) “Adoption of agile approaches to large IT projects” and Kalenda (2017) “Scaling agile software development in large organizations”.

In the thesis “Analysis of agile software development methodology usage in work teams” Horváthová (2019) investigates the impact of the Scrum practices on a case company. As part of the work, a software tool is developed and presented. The implemented tool helps to process the reflections on estimations for team members. The tool is developed using Python programming language. It extracts data from a GitLab project, process it and represent the data in a graphical form using a spreadsheet in Google Sheets.

1.3 Specialized literature

Considering the main goal of my Master thesis - analysis, design and implementation of web application, I decided to include to my research also a literature, which is dedicated to software design and development. To obtain the results I used Union Catalogue of the University of Economics, Prague. As I mention in Introduction section of my work, this Master thesis does not aim to describe and compare individual project management methodologies or to cover all agile software development principles and practices, therefore I did not include books about description or evaluation of agile principles, practices and frameworks to the final results. The outcome of the literate review is presented in Table 3.

Table 3 Review results: Specialized literature

Name of the publication	Author	Year	Description of the publication
Object-oriented analysis & design.	Mathiassen, Lars et al.	2000	The book shows and explains the techniques such as problem-domain modeling based on class diagrams, application-domain modeling rested on use cases and functions. The book propagates a design, which is based on layered architecture and encapsulation.
Příklady modelů analýzy a návrhu aplikace v UML.	Buchalceková, A., Stanovská, I.	2013	The publication is a student book in Czech language, which describes how to model and design information systems using the Unified Modeling Language (UML).

SignalR: real-time application development.	Ingebrigtsen, Einar.	2015	The book is a fast-paced and example-oriented guide to learn the fundamentals of SignalR framework to develop, test, and deliver real-time applications.
---	----------------------	------	--

“Object-oriented analysis & design” (Mathiassen, 2000) is a book, that is often referenced and used in universities in system development courses, as it includes tested and mainstream analysis and design practices. The book provides exercises, teaches design patterns and guides to design of high-quality user interfaces. The author shows how to carry an object-oriented software design from use cases all the way to coding and testing. The book demonstrates common analysis and design mistakes, shows how to catch and fix them, and makes recommendations how to avoid making the same errors in the future.

“Příklady modelů analýzy a návrhu aplikace v UML” (Buchalceková and Stanovská, 2013) is a student book in Czech language, which offers practical methods for mapping business processes and estimating the complexity of the IT solutions. The publication provides models and recommendations, how to create a documentation using the Unified Modeling Language.

The “SignalR: real-time application development” publication (Ingebrigtsen, 2005) is a step-by-step guide, that gives practical advices how to build from scratch and test real-time web applications. The publication helps to gain an understanding how the SignalR framework works.

2 Research plan and Methodology

To fulfill the main goal of my Master thesis I used Design Science Research (DSR) methodology. Design science research typically follows a goal to improve the current state of practice instead of explaining and making sense of existing one (Iivari and Venable, 2009). Iivari and Venable define DSR as “a research activity that invents or builds new, innovative artifacts for solving problems or achieving improvements” (2009, p. 4). A result of the DSR is a purposeful artifact, that “*can be a product or a process; it can be a technology, a tool, a methodology, a technique, a procedure or a combination of any of these for achieving some purpose*” (Venable and Baskerville, 2012, p. 142).

The DSR process involves six activities (Peppers et al., 2007):

- 1) Identification of the problem
- 2) Definition of objectives
- 3) Design and development
- 4) Demonstration
- 5) Evaluation
- 6) Communication

To conduct DSR I am using a case study project. Case studies are one of the methods proposed by Johannesson and Perjons in their work “A Method Framework for DSR” (2014). This framework provides a description of methods, how the individual DSR activities can be carried out.

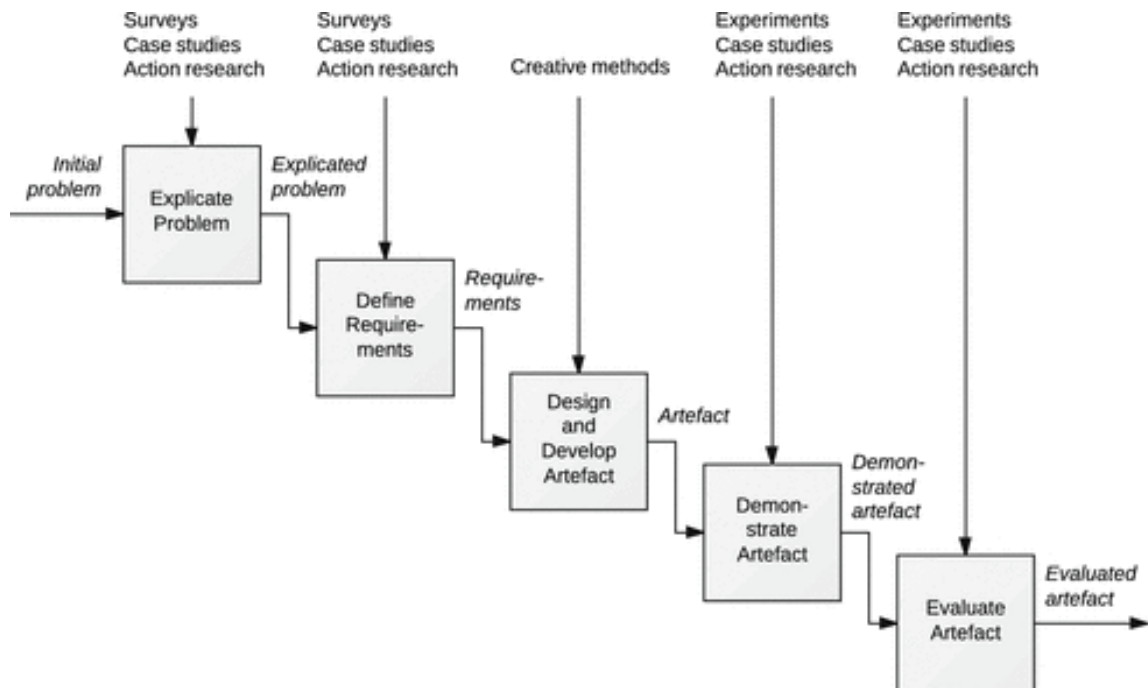


Figure 1 Method Framework for Design Science Research (Johannesson, P. and Perjons, E., 2014)

Identification of the problem and motivation

The current state of the agile practices, which are realized on the case study project, is not satisfying for team members. The inconveniences and issues are directly connected to the project's main characteristics – large team size and geological dispersion. The agile processes, which are carried on the case project, assume face-to-face communication and interactions, especially on the planning or review meetings of the team. Carrying meetings at only one site isolates the rest of the team members, and the feeling of united team wanes. The lack of perception coworkers as the part of the team has influence on reaching a common goal. The results of systematic literature review, conducted in the chapter one, show, that these problems are common for virtual teams and global software development in total. Just as the number of companies, which are adopting agile methodologies, grows each year, the percentage of companies with GSD grows as well. Therefore, the author can assume that more and more companies will encounter such problems.

Definition of objectives for a solution

The main objective of this paperwork is to develop an artifact – a prototype of web application. The application should support agile project management practices and improve agile processes in virtual teams. The prototype is developed on demand for the case project and its team, who is engaged in a bank environment. However, the web application is not limited to the case project or financial environment and can be used in different industries, not only in a software development field. As the article “Virtual by Design” (Lous et al., 2018) states, a careful selection of the practice-tool combinations increases the effectiveness and facilitates each team member to fully participate in processes regardless of the different physical location. The main purpose of my paperwork is achieved by completing the individual smaller goals such as conducting a current state analysis, establishing requirements, providing a detailed design and a development process. The solution is tested and validated by the case project team.

Design and development

Based on the defined goals, the analysis of current state on the case project and gathered requirements from the client, three agile activities are decided to include in the application prototype. The prototype includes the design for Sprint Retrospective, Poker Planning and Brainstorming sessions. Considering the scope of this Master thesis and time restrictions, the physical development, further demonstration and evaluation phases were reduced to Sprint Retrospective and Poker Planning features. For the prototype implementation the programming language C# was used together with MVC ASP.NET and SignalR frameworks. For object-relational mapping (ORM) and communication with a database was used Entity Framework. The development of physical prototype is used for testing of selected technologies and to solve defined problems on the case project. This experience should assist in the further implementation of final application.

Demonstration

After the development and final confirmation with the client the prototype was published on the Microsoft Azure, cloud computing platform. The prototype can be found on the web address <http://cpsagile.azurewebsites.net/>. The prototype was then demonstrated to the case project team and used to perform Sprint Retrospective and Poker Planning Scrum ceremonies.

Evaluation

The implemented web application was presented to the team and used for Retrospective and planning meetings. At the end of the meetings, the case project team had made comments and provided recommendations about new tool. The comments were noted and planned for the implementation in future versions of the web application. The list of future improvements is placed in the final chapter of this paper. Overall team impression was positive and the new way of carrying planning and Retrospective meetings were met as the improvement of current processes.

Communication

The web application is open source, and the programming code is stored in the public repository on GitHub - collaboration platform for software developers. This Master thesis will be available on Thesis.cz portal and in the internal system of University of Economics in Prague. The word about the application and new way of carrying planning meetings will be spread among other teams that are working on the case project.

3 Agile project management

In this chapter I would like to shortly introduce the differences between agile and traditional methodologies. As it states in the research article “Project Management Tools for Agile Teams” (Mihalache, 2017), the tools for agile practices should be fundamentally different from the traditional tools due to dynamic and adaptive nature of the agile processes. I believe that making such comparison will help better understand the current needs of the client. To the agile project management methodologies belong such methodologies as Scrum, Kanban, Lean Software Development, XP, Crystal, Dynamic Systems Development Method (DSDM) and Feature Driven Development (FDD). Based on the data from the latest State of Agile Report, provided by VersionOne company (CollabNet VersionOne, 2019), Scrum remains the most widely practiced agile methodology. The client applies Scrum methodology on the case project, for that reason the second part of this chapter is dedicated to Scrum framework, its roles, artifacts and ceremonies.

3.1 Comparison of traditional and agile methodologies

The cornerstone of agile methodologies is the Agile Manifesto, which was written and signed in 2001 by seventeen independent-minded software practitioners (Agilemanifesto.org, 2001). The Agile Manifesto declares the following values:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

The manifesto asserts, that while there is value in the items on the right, the items on the left should value more. From the manifesto is apparent how traditional and agile approaches differ in significant ways. The core differences are presented in Table 4.

Table 4 Traditional versus agile software development (Nerur, Mahapatra and Mangalaraj, 2005)

	Traditional	Agile
Management Style	Command-and-control	Leadership-and-collaboration
Control	Process centric	People centric
Development Model	Life cycle model (Waterfall, Spiral, or some variation)	Product delivered incrementally with short term, iterative development cycles

Project Cycle	Guided by tasks or activities	Guided by product features
Team Composition	Definitive team roles with individual accountability — favors specialization	Self-organizing teams, cross-functional roles with the team sharing accountability equally
Communication	Formal	Informal
Desired Organizational Structure	Mechanistic (bureaucratic with high formalization)	Organic (flexible)
Client involvement	Important. Stakeholders are typically involved only at the beginning and end of project life cycle	Critical. Stakeholders are deeply involved throughout project development
Scope and Resource Management	Fixed scope with estimated resources and time	Fixed resources and time with estimated scope

Traditional methodologies put great emphasis on the precise planning, formal documentation and review process. The differences in scope and resource management is illustrated in Figure 2. In the traditional approach requirements are negotiated with the stakeholder and thoroughly documented in the project planning phase in order to set a fixed scope of work. Resources and time are then estimated based on the fixed scope. In traditional methodologies stakeholder involvement is important only during requirements and acceptance testing phases. Agile processes, on the contrary, are people centric and value driven. With fixed resources and time frame, the scope becomes most negotiable and therefore enables a quickly respond to change. Agile approach requires a continuous, iterative prioritization of scope, collaboration with the customer and his continuous involvement in a process.

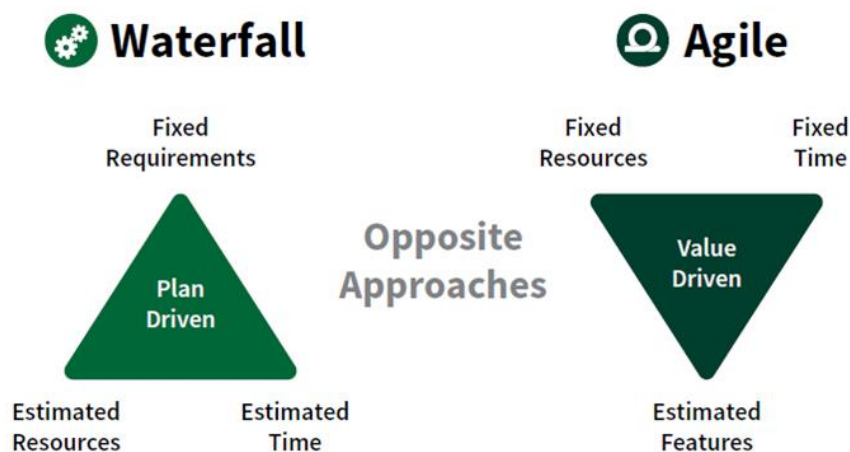


Figure 2 Scope, resources and time in traditional versus agile approach (Ca.gov, 2017)

3.2 State of Agile report

In this section I would like to communicate the current state on worldwide adoption of agile by presenting the results of the 13th annual State of Agile survey. The State of Agile is the longest running report on agile software development, which is conducted by VersionOne (CollabNet VersionOne, 2019). The report explores the usage of agile practices and the adoption and scaling of agile methodologies. The survey investigates both the benefits of the adoption and challenges that are faced by organizations. In the 13th State of Agile survey were collected 1319 responses from individuals in the global software development community. 97 % of respondents says their organization practices agile development methods. Nevertheless, only 22 % says that all teams in the organization are agile and 48 % claims that less than a half of teams are agile. The number of distributed teams, which are practicing agile methodologies, is growing. In 2012 only 35 % of participants had stated, that their organization had agile virtual teams (CollabNet VersionOne, 2013). From 2012 to 2014, the percentage of respondents who had agile distributed teams jumped from 35 % to 80 % (CollabNet VersionOne, 2015). The report shows that Scrum (54 %) and hybrids of Scrum (32 %) continue to be the most widely used agile methodologies.

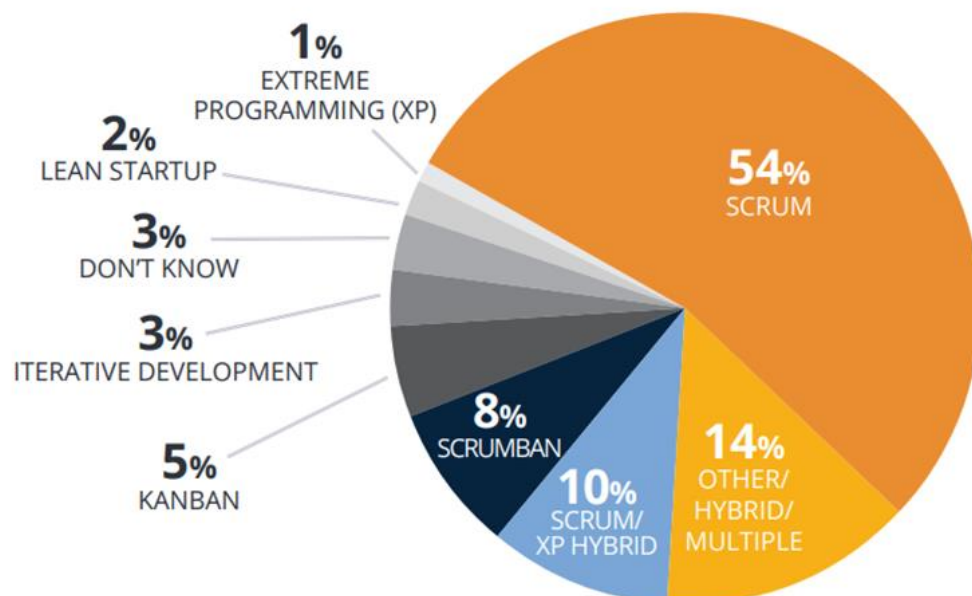


Figure 3 Agile Methodologies used by respondents (CollabNet VersionOne, 2019)

3.3 Scrum framework

Scrum is an agile project management framework, which is widely used by software development teams. The Scrum methodology follows the common values and principles of agile, but introduces additional artifacts and practices, which helps day-to-day interactions and activities of the team. Scrum framework was created and defined by Ken Schwaber and Jeff Sutherland, two of the 17 initial signatories of the Agile Manifesto. In 2002 they founded the Scrum Alliance® and set up the Certified Scrum accreditation (Scrumalliance.org, 2019). In 2009 Scrum.org was founded and an official public document The Scrum Guide™ was published by Schwaber and Sutherland (Scrum.org, 2019). Five Scrum values – Courage, Focus, Commitment, Respect and Openness are foundations of three framework main pillars – Transparency of processes, Inspection and Adaptation of practices.



Figure 4 Scrum Values (Scrum.org, 2019)

Scrum is the most popular agile methodology according to the 13th annual State of Agile report. While the Scrum methodology is most frequently used by software development teams, its principles and lessons can be applied to all kinds of teamwork. Scrum is defined by team roles, artifacts and events (ceremonies) that in combination help teams to structure and manage their work. The whole section and subsections below are based on information from the official Scrum Guide, which is available on Scrum.org (Scrum.org, 2019).

3.3.1 Roles

The main functional units in the framework are Scrum teams. Scrum teams are self-organizing and cross-functional. Team autonomously decides how to accomplish a result, there is no team leader to delegate tasks or choose how a problem will be addressed. Each team member is continuously involved in the process and is expected to carry a product from initiation to completion. There are three key roles in a Scrum team:

- The Product Owner
- The Development Team
- The Scrum Master

The Product Owner serves as a communication bridge between stakeholders and a team – he represents the stakeholders and customers, but also represents a team to the overall stakeholder community. The Product Owner is accountable for maximizing the business values that the team delivers; therefore, he defines and prioritizes work items in customer-centric terms. This role should be occupied only by one person and should not be combined with the Scrum Master.

The Scrum Master is accountable for managing the Scrum Process and removing any possible impediments and barriers that arise during the project. The Scrum Master does not manage the Development Team, because the Team must be self-motivated and self-organized enough to deliver require work without controlling. The Scrum Master is responsible for guiding the Team and creating an appropriate work environment. The Scrum Master acts as a coach, helping the Team to follow Scrum values and build a product according to the Scrum framework.

The Development Team is dedicated to creating and delivering the products or services. It is a cross-functional, self-organizing team with a very high degree of autonomy and accountability. The autonomy is a core of the Scrum framework. In Development Team decisions are made collectively and everyone is in charge of figuring out how to achieve desired results, no team leader can be appointed. The Product Owner sets the priorities, the work is guided by Scrum Master, but all other responsibilities are laid on the Development Team. For instance, the Team decides how many items to build during an iteration, and how best to accomplish that goal.

3.3.2 Artifacts

Artifacts in Scrum represent work or value, which maximizes understanding and transparency of key information about the product under development, the activities being planned, and the activities done in the project. The main goals of artifacts are to maximize transparency and to promote a shared understanding of work. The Scrum framework describes three primary artifacts: the Product Backlog, the Sprint Backlog, and the Product Increment.

The Product Backlog is the ordered list of work to be done by the Development Team. It lists all features, requirements, enhancements and fixes that are to be made in the feature releases. The Product Backlog is the single source of requirements and is owned by

the Product Owner. Having a single ordered list makes organizational priorities transparent to all internal stakeholders. The list is visible to everyone, but only the Product Owner can make changes to it. The Product Owner is responsible for ordering the Product Backlog and filling its content. The single Product Owner responsibility provides transparency to the organization about the decision-making process used to prioritize the work. The Product Backlog is a dynamic artifact, it is constantly updated based on a current knowledge of reality. This attribute provides an ability to quickly react on changes in a business environment and prioritize the work according to current needs. The Product Owner collaborates with the Development Team to constantly refine the Product Backlog. The dynamic, ongoing and collaborative refinement promotes a shared understanding of the work.

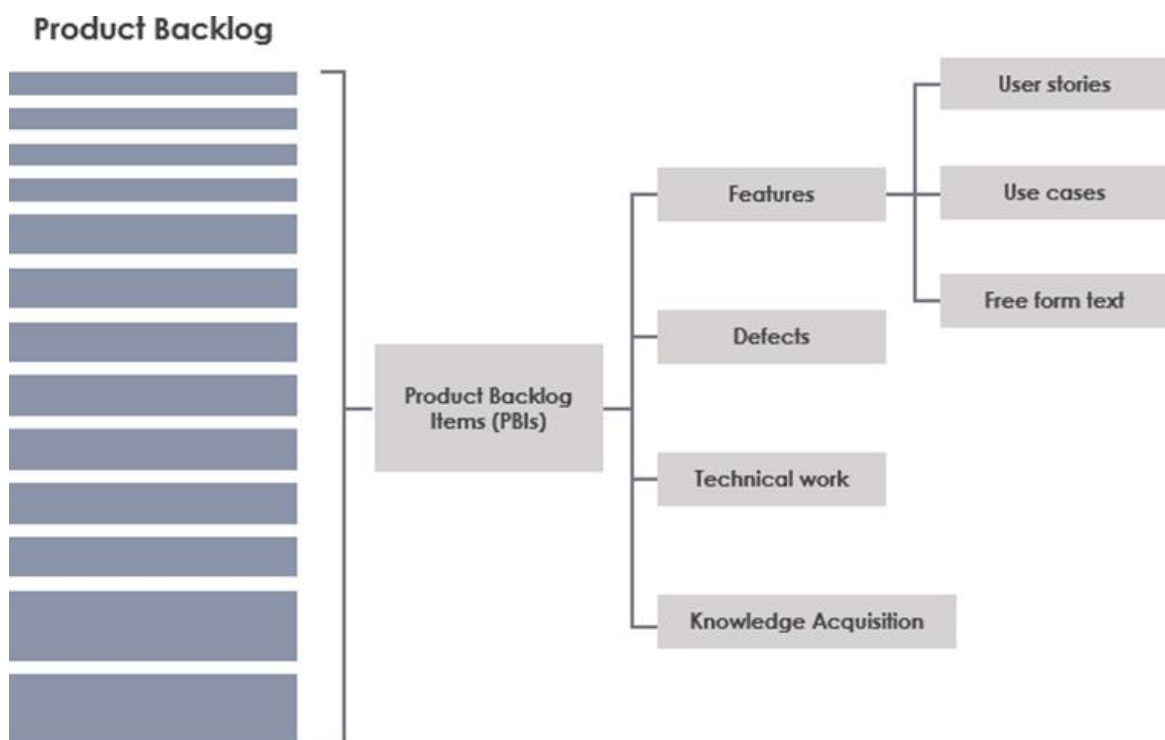


Figure 5 Product Backlog (Visual-paradigm.com, 2018)

The Sprint Backlog is collaboratively created during Sprint Planning by the Development Team. It is a plan how the Team is going to deliver the Sprint Goal and related the Product Backlog items during the current Sprint. The Sprint Backlog is a transparent real-time picture of the work in the current Sprint. Only the Development Team can change the Sprint Backlog during a Sprint.

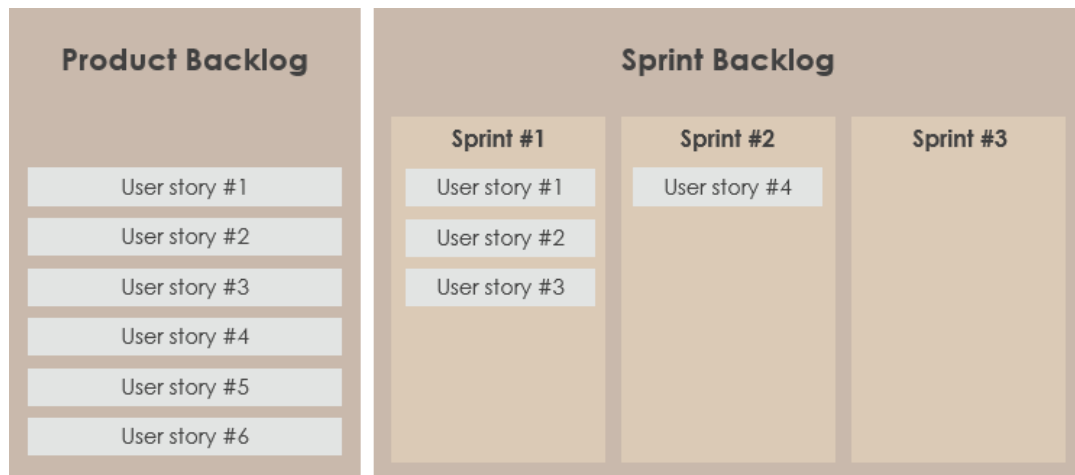


Figure 6 Sprint Backlog (Visual-paradigm.com, 2018)

The third and last Scrum artifact is the Product Increment. Official Scrum Guide defines the Increment as an output that is formed from all completed Sprint Backlog items during current Sprint plus the value of all previous Increments. At the end of a Sprint, the new Increment must be complete according to the Scrum Team's definition of "Done". The Increment provides an accurate, transparent picture of the state of the product at the end of each Sprint. It helps to implement one of twelve agile principles - *"working software is the primary measure of progress"* (Agile Alliance, 2001). Since the Increment is a complete working piece of software, it is usable by stakeholders and provides an opportunity for inspection and feedback. The Scrum artifacts contribute to transparency and shared understanding, therefore good decisions can be made through inspecting and adapting.

3.3.3 Ceremonies

Scrum is executed in so called Sprints. A Sprint is a short iteration of work usually lasting one month or less, at the end of which Increment is created. In order to maintain transparency and effective communication during a Sprint, Scrum ceremonies are held to ensure proper execution. Scrum events are meetings that are unique to Scrum Teams – Sprint Planning, Daily Scrum, Sprint Review and Sprint Retrospective. These meetings ensure that the Scrum Master, the Product Owner and the Development Team are all in-sync.

A Sprint Planning meeting is held prior to upcoming Sprint. This ceremony requires the participation of all the Scrum Roles: the Product Owner, the Development Team and the Scrum Master. The Scrum Team holds a Sprint Planning meeting in order to establish a Sprint Goal and choose, which Product Backlog items will be implemented during the Sprint. The Product Owner prepares a prioritized list of Product Backlog items and presents it to the team. Together, the Scrum Team estimate how much time and effort will take to complete the items on the list. The estimated items are usually presented in a form of User Story.

User Story is a short, simple description of the desired feature, which is told from the user or customer perspective. User Stories usually follow a simple template:

As a < type of user >, I want < some goal > so that < some reason >.

An example of User Story can be – “*As a moderator I want to be able to export the results in an Excel file so that I can share it with other stakeholders*”. After the estimation, the Team chooses what User Stories will be included in a Sprint Backlog and developed in an upcoming Sprint. One of the possible ways how to carry a Sprint Planning meeting is a Poker Planning. Poker Planning is consensus-based estimating technique. Each estimator has a deck of cards with numbers on them. A typical deck for Poker Planning consists of cards showing The Fibonacci sequence – 0, 1, 2, 3, 5, 8, 13, 21, 34, 55 and 89. The values on the cards represent the number of story points or other units in which the team estimates the feature. The Product Owner provides a short description of one User Story and the estimators discuss the feature. Then each estimator privately picks one card with the number representing an estimation. When everyone is ready, all cards are revealed at the same time. If all estimators picked the same card, then this value becomes the estimate. Otherwise, the estimators discuss their choices and the estimation process is repeated until a consensus is achieved.

Another Scrum activity is a Daily Scrum. Once a Sprint began, short Stand-up meetings are carried every day. Daily Scrum is typically a 15-minute stand-up meeting to ensure transparency across the team. Every participant of the event answers three questions: what they completed yesterday, what they are working on today, and if there is something that blocks their work. Daily Scrum improves communication and helps to synchronize work inside the team.

A Sprint Review is a Scrum ceremony that is held at the end of the Sprint. The goal of this meeting is to inspect the Increment and adapt the Product Backlog. The Sprint Review includes the Scrum Team and key stakeholders. The Development Team demonstrates the Increment, often in a form of Demo or showcase. Each team member reviews the newly developed features and answers the questions. Sprint Review is a great opportunity to provide and get feedback from the stakeholders. The participants review timeline, resources, marketplace, and collaborate on what is the most valuable thing to do for the next Sprint. The result of the Sprint Review is a refinement of Product Backlog, feedback from stakeholders and a shared understanding of common goals.

The fourth unique Scrum ceremony is a Sprint Retrospective meeting. The aim of Sprint Retrospective is to reflect on the previous Sprint and to make sure that the product and development culture are constantly improving. The meeting occurs at the end of the Sprint and includes the Development Team, the Scrum Master and the Product Owner. The Retrospective meeting is a way to figure out what had been done well, what did not go as planned, and what can be improved. The Sprint Retrospective meeting facilitates the team to focus on its overall performance and identify strategies for continuous improvement.

Scrum events provide an opportunity to inspect and adapt processes. The ceremonies help the team to maximize their productivity, maintain transparency and enrich communication. The Scrum framework with its main ceremonies, roles and artifacts is illustrated in Figure 7.

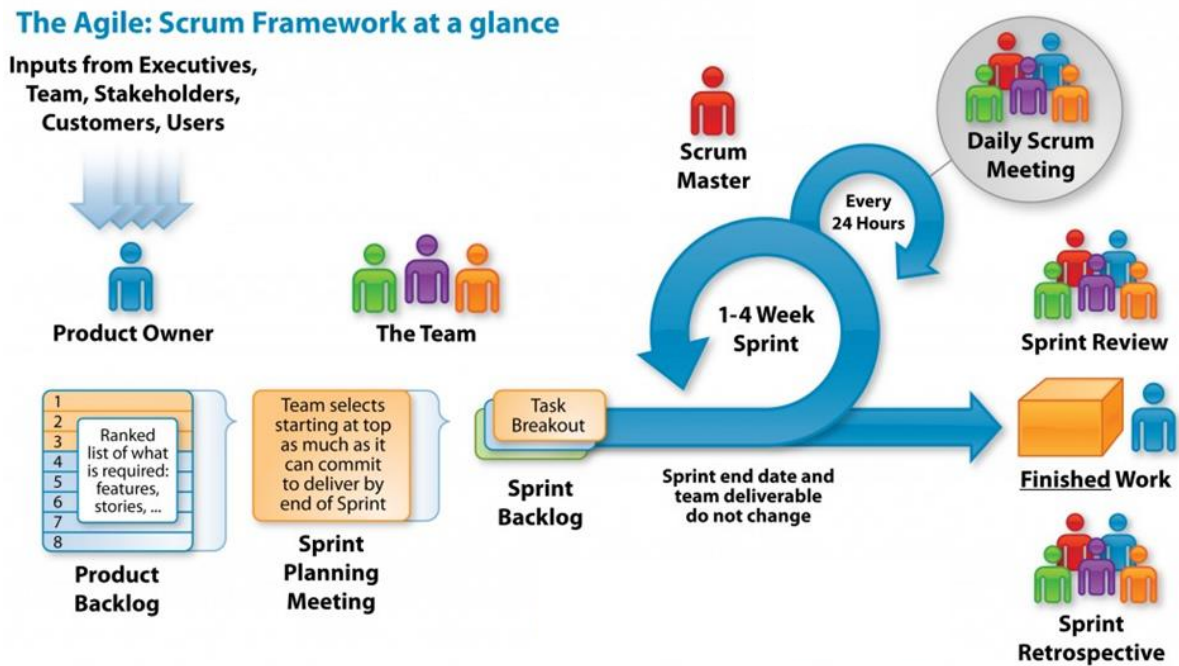


Figure 7 Scrum framework at a glance (Neon Rain Interactive, 2015)

3.4 Summary

In this chapter I pointed the basic differences between traditional and agile methodologies, as it is crucial to understand the fundamentals of agile principles, in order to create a web application, which supports such practices. I also described the Scrum framework, its roles, artifacts and events. The Scrum framework is used on the case project; therefore, the next chapter is devoted to the investigation and specification of the current agile practices on the case project.

4 Current agile practices on the case project

This chapter is dedicated to the description of the current Scrum processes on the case project. Before starting analysis and designing a prototype of the application, it was necessary to understand the processes of current practices on the customer side. In order to get deeper and find out the customer needs I participated in different Scrum activities together with customer's agile team. After a comprehension of weaknesses and disadvantages of the current solution I am looking into several existing competing web applications that address similar tasks, then I am making comparison and evaluation of these products in order to get a full view of the problem area.

4.1 Current client's solution

This section describes the agile processes which are currently practiced on the case project. At the end of the section I summarize the main obstacles and limitations of the present agile practices.

On the case project are currently used such Scrum ceremonies as Sprint Planning, Daily Scrum, Sprint Review and Sprint Retrospective.

Daily Scrum is carried out by daily stand-ups where each member of the Development Team is sharing their status update by briefly answering three questions:

- *What did I do yesterday?*
- *What will I do today?*
- *Are there any impediments on my way?*

To communicate with remote members of the team during stand-up are used such enterprise software solutions as Skype for Business² or Google Hangouts³.

Sprint Review is holding each second Thursday and is called "Demo Thursday". Within the hour the Development Team demonstrates to the Product Owner and other stakeholders all the work that has been completed since the last demo. Since there is no need for special agile tool, I will not address the topic of Sprint Review further in my paper.

As a part of the Sprint Planning the team is practicing a Scrum poker activity. All members of Scrum team attend planning meeting. During the Scrum Poker activity, the Development Team is operating with several copies of a deck of cards. Planning Poker with a real paper

² Previously was used Lync, which was renamed to Skype for Business in 2015 and replaced by Microsoft.

³ Google Hangouts is a communication platform which supports messaging, video chat, SMS and VOIP features.

deck of cards requires that all members of the team were present in one room; therefore, it creates a limitation for virtual teams. This limitation is currently solved in such way, that the Scrum Master creates Skype meeting and sends Excel file with User Story list to all team members. Remote members of the team join Skype meeting and share their estimates by writing the exact number in Skype at the same time when all cards are being revealed. Occasionally this process can be extremely inconvenient due to different internal and external reasons such as quality of Skype connection, or problems, which are raised due to impersonal communication. The remote members of the team do not clearly see estimation of other team members, as the cards are laying on the physical table. For that reason, the Scrum Master reads other estimations to the remote participants through Skype. After a User Story is estimated, the Scrum Master is writing the final estimation to Excel file, in order to save the results of estimation.

Participating in the Sprint Retrospective activities was the most interesting part for me, because it was pointed and defined from the beginning as the weakest supported agile process on the case project. In this process I found several limitations and weaknesses straight away. At the present time a whiteboard, markers and colorful sticky notes are utilized for Retrospective activities. The Sprint Retrospective event is held as a start-stop-continue meeting. Using this approach each team member is identifying specific things that the team should:

- *Start doing*
- *Stop doing*
- *Continue doing*

Retrospective event is held gradually and can be divided into three phases. First phase is the preparation of points, which usually lasts from five to fifteen minutes. Participants of the event are writing their ideas privately on the sticky notes. When everyone is ready, each member of the team presents their notes, by placing a sticky note on a whiteboard under the corresponding column gradually one by one. When a note is placed on the whiteboard, the owner of the note makes comments on it and the team discusses the idea. The Scrum Master moderates the Retrospective event. The disadvantage of this approach in the virtual team is that a moderator of the meeting, mainly a Scrum Master, needs at first to rewrite the ideas of the remote team members to sticky notes, and then at the end of the meeting, collect all sticky notes and rewrite it into electronic form. Another problem lies in the poor visibility of the text on the sticky notes, which are adhered on the whiteboard, especially for remote users, who are connected over standard communication platforms such as Skype.

Although the Scrum framework sets, that the entire team, including both the Scrum Master and the Product Owner should participate in the Scrum Retrospective meeting, at the Retrospective meeting on the case project is currently presented only the development team with the Scrum Master. The Product Owner does not get involved in this activity mainly because of the impossibility of the actual presence and other obstacles, which were described above. By fulfilling the main goal of this Master thesis, creating a web application and transforming certain agile processes to the digital shape, such hindrances

will be overcome, the regular problems of the virtual teams will cease and an overall improvement of agile processes on the case project is expected.

The Brainstorming as another agile activity is likewise implicated in the client's project management processes. Brainstorming is a thorough and detailed event with a specific aim to find a solution to a problem. During the event, all members of the agile team are invited to come up with as many ideas as possible to resolve the issue. As the result of Brainstorming session, several of the most acceptable options are picked to solve the problem.

4.2 Competing ready-made solutions

In this section I am examining and evaluating existing analogous solutions from other suppliers, which address the identical problem topic. The main objects of my investigation were web applications with similar features and on a cost-free basis. This means that they must have free access and support at least one of the following agile techniques – Retrospective, Poker Planning or Brainstorming.

4.2.1 RealtimeBoard

RealtimeBoard is an online application, which can be found on <https://realtimeboard.com> address. It provides a large scale of different independent features, such as creating of User Story Map, Mind Map, Kanban or Scrum Board, Retrospective, Flowchart and Product Roadmap. I examined tools, which are designed for Retrospective session and found several strengths and weaknesses of this solution.

As it can be seen from Figure 8, the main advantage of this web application is a bright colorful interface, which is well visible, captivating and aesthetically pleasing.

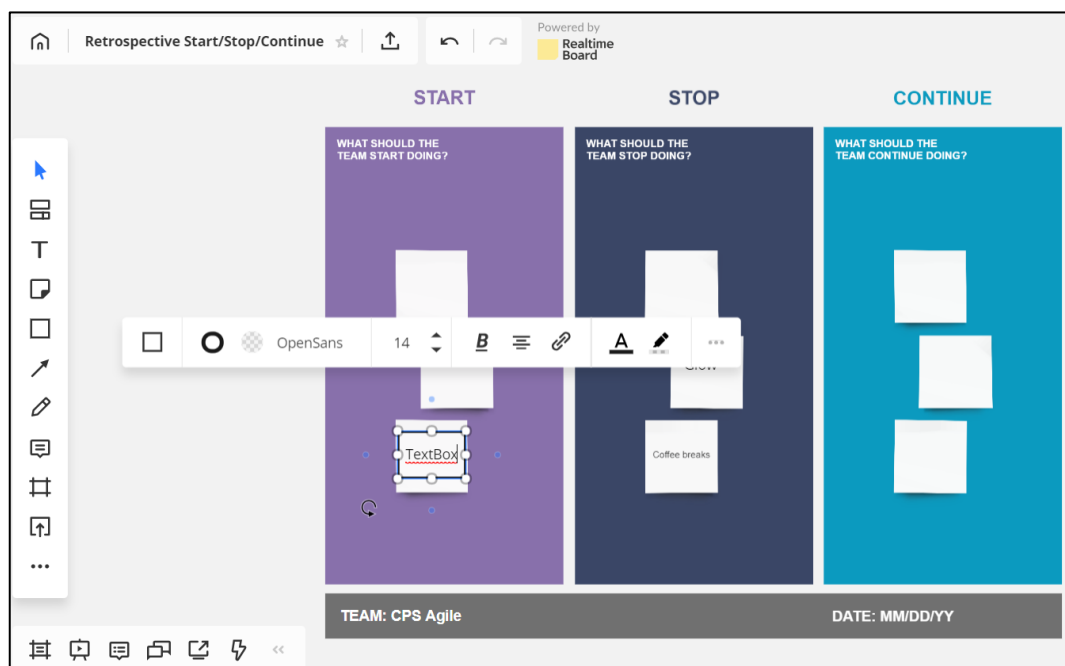


Figure 8 Retrospective feature of RealtimeBoard. Available from <https://realtimeboard.com>

This retrospective solution of RealtimeBoard is made as a background picture, where user can put a text into a text box. Therefore, it is lacking usability as it is extremely inconvenient to add text in a text box, and most of the time was spent by editing and adjusting of cells to make sure it fits boundaries and has appropriate font size. RealtimeBoard is probably a good solution for building Flowchart and User Story Map, but not for a Retrospective event, as it distracts attention from the core activity.

4.2.2 IdeaBoardz

The next online tool for agile activities I found interesting for the evaluation is IdeaBoardz. IdeaBoardz is a team collaboration tool, which is free of charge and is available on <http://www.ideaboardz.com>. It allows teams to collectively brainstorm and retrospect. One of the features of this application is an ability to create several alternatives of Retrospective with different formats and structures, for instance, a star fish retrospective can be created. A user of IdeaBoardz can additionally choose and set amount of sections he wants to use and then write down a name for each section title. Other coworkers can be invited to participate in a created activity simply by sharing a link to current event. Then all users, who have a link, can add their thoughts below each column by creating a virtual sticker note. Information in columns are synchronized and shared among all users with a very little time delay. This web application offers a feature to vote for each virtual note below columns and besides there is a possibility to export results in PDF or Excel formats.

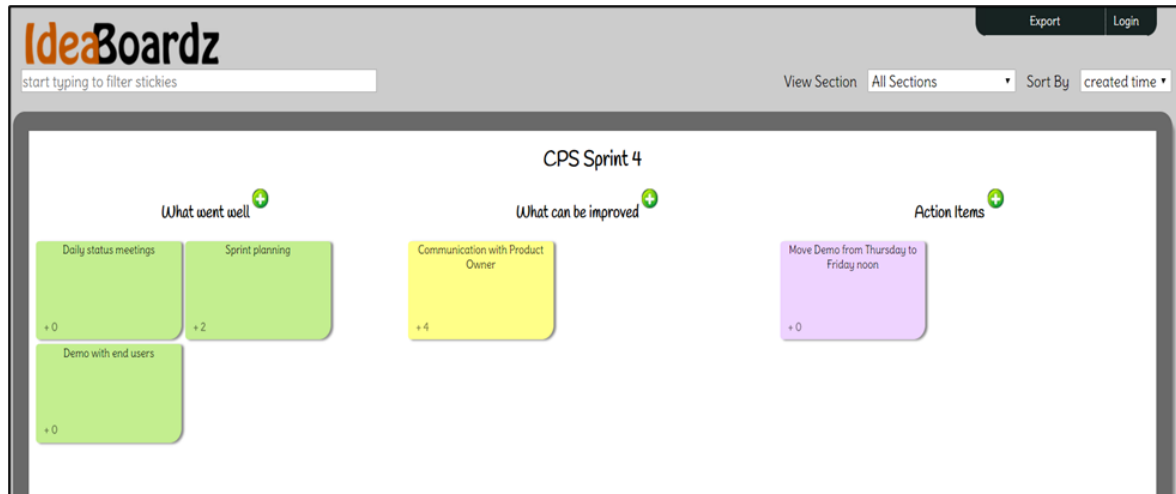


Figure 9 Retrospective feature of IdeaBoardz. Available from <http://www.ideaboardz.com>

Advantages:

- Ability to export results
- User-friendly interface
- No need for a registration, user can join a created event by clicking on shared link

Disadvantages:

- The event itself does not take place in stages, but everything is instantly shared among users
- Information about current participants of the event is not displayed, thereby there is no chance to find out what users and how many of them are taking part in the event

4.2.3 Pointing Poker

Pointing Poker is online web application, which can be found on <https://www.pointingpoker.com/>. This tool is created for virtual and co-located agile teams for their planning sessions and enables to effectively communicate points for stories. While creating a Pointing session, user can set values (points) for cards, which will be used in a planning session. This ability helps to adapt a planning session for team's needs in current Sprint. The good thing is that information about logged-in users is displayed during the session and information about who has voted is tracked. Unfortunately, there are no roles, so every participant of the event has the same access rights and can edit a user story description and reset votes for any user story. During a voting process the user selects the appropriate one card with a number represented by points and only after his/her selection the votes of others are displayed. The Pointing Poker application also enables to carry out a Retrospective session, which takes place in two phases. The first phase is called "Entry Mode", where users add their notes to corresponding columns, without seeing alternative notes from other users. The second phase is called "Review", where all at once everyone can see all notes. Every participant of the event can switch between these two phases, which is caused again by lack of roles and inability to set access rights.

The screenshot shows the Pointing Poker application interface for a session titled "CPS Agile team" with Session ID 4670. The story description is "develop a loans processing web service". There are buttons for "Clear Votes" and "Show Votes". A grid of point values is displayed: 0 points, 1/2 point, 1 point, 3 points, 2 points, 5 points, 8 points, 13 points, 20 points, 40 points, 100 points, and ?. The "Statistics" box shows "Time taken: 0:00:41", "Average: 8", and "Points" and "Votes" both at 8, with a "Consensus!" message. A "Player" list shows "CPS Agile team" with 8 points. A link to invite others is provided: <http://www.pointingpoker.com/4670>. A text box for entering email addresses for invitees is at the bottom. A speech bubble says "You look lonely" and "Invite a teammate!".

Figure 10 An estimating session from the Pointing Poker application. Available from <https://www.pointingpoker.com/>

Advantages:

- Creator of a pointing session can set values for voting cards
- Information about logged-in users is displayed during a whole session

Disadvantages:

- Inability to export results
- Lack of pre-set roles and access rights
- Inconvenient and outdated design

4.2.4 Fire Poker

Web application Fire Poker, which is accessible from <https://firepoker.io/>, has a lot of possible settings and therefore provides a great autonomy for users and adjusts a solution to team's demand. This application does not lack roles. The first member logs in as a moderator, sets an event and adds user stories. Then the moderator can accept the voting result, cancel the vote or repeat a whole process of voting again. A list of connected participants and their chosen cards are displayed during an event. The selected cards of other users are revealed to each participant immediately after he finishes his/her voting. The number of voting rounds is not limited, and user story can be modified by anyone.

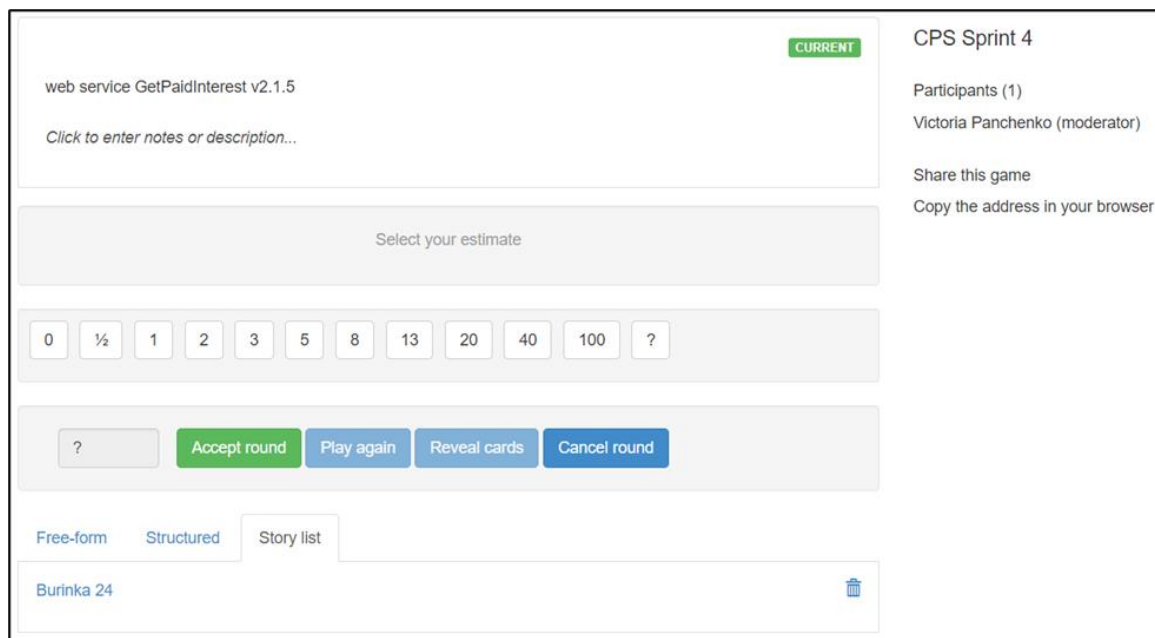


Figure 11 Fire Poker application. Available from <https://www.pointingpoker.com/>

Advantages:

- Ability to set values for voting cards
- List of currently logged-in users

Disadvantages:

- Poorly arranged and confused user story setting menu
- User story can be modified by anyone

4.3 Summary

A comparison of the existing competing ready-made solutions showed that the vast majority of applications that have been reviewed are unsatisfactory and unsuitable for the client's needs. Reviewed and tested web applications are usually short of voting option or capability to export results from an application. The most cherished thing that the applications do not offer is the possibility of carrying Retrospective gradually step by step by a moderator. Having an export option, an option for voting phase for Retrospective event and option for managing phases of the events are vital points to the team. These are the main reasons, why there is a demand for such web application from the client's side and why a final product could easily succeed among a number of competitors.

5 Analysis and requirements' establishment

Based on the analysis of current client's solution and available products on the market, which was held in the previous chapter, I am discussing the basic needs and establishing requirements from the client in this chapter. Determination and clarification of requirements is perceived as the most important part of the whole process because it is a vital document that client must agree with a supplier. In a case of any ambiguity during the development phase or when the product is implemented and delivered, the main document to refer and compare is the document where requirements are declared and agreed by both sides. Therefore, the whole development process later is mainly based on the objective to meet the requirements which are agreed upon in this chapter.

The chapter first describes the requirements on the application, which were agreed at the initial negotiation with the contracting authority. Further I am describing the general requirements which have been agreed upon later with the client. During the process of gathering information I as well held a few meetings with other project managers and Scrum Masters, who are engaging in an agile project delivery. I made a few interviews with them in order to better understand their needs and reasons for developing such product. One section of this chapter is in addition dedicated to evaluation and discussion whether the desired application is suitable for the usage in other fields besides IT/ICT area. At the end of the chapter I am summarizing the final list of functional requirements. The whole process of definition and establishment of the requirements is reflected in the following Figure 12 Requirements establishment process.

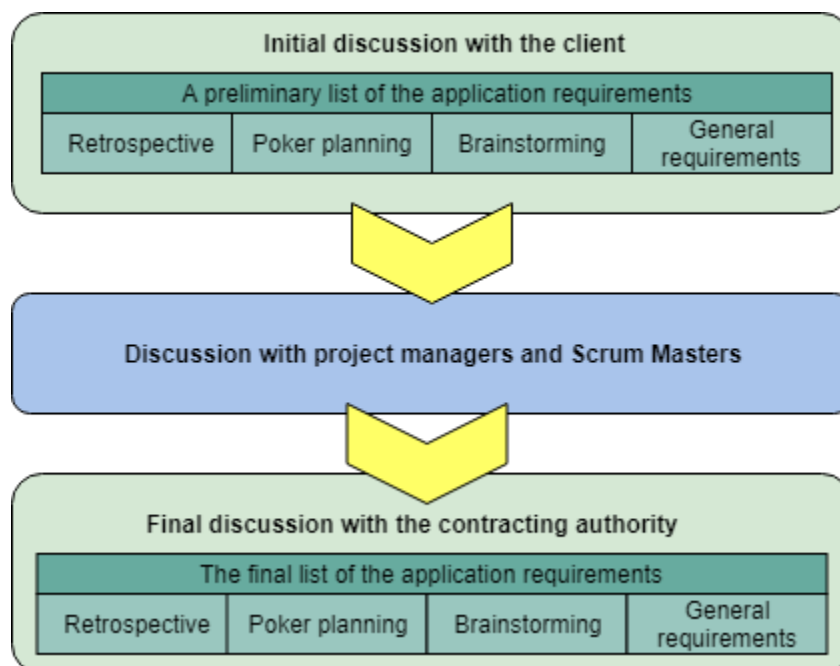


Figure 12 Requirements establishment process

5.1 Initial discussion with the client

The initial discussion with the client ran as a kickoff meeting, where the first meeting with the supplier and the client took place. As a result of the kickoff meeting with the contracting authority was established a preliminary list of the application requirements. During the initial discussion the name of the project was chosen. The client determined that the final product, a web application, will be called “CPS Agile”. The main focus of the “CPS Agile” web application should be concentrated on an assistance and support of the agile development in a large distributed team. The application should include such tools, which assure several major agile development techniques such as:

- Poker Planning
- Retrospective
- Brainstorming

In addition to the main features of web application, the client set general requirements on the “CPS Agile” application, which have been likewise agreed on the kickoff meeting and which are described in the next subsection of my paperwork.

5.1.1 General requirements on the application

During the initial meeting the general requirements on the application were discussed with the contracting authority. The client desired that the “CPS Agile” application would be accessible through commonly used internet browsers and the whole application would run in English language. The application should be able to run in two modes, which depends on the chosen user type from the possible two user roles – a moderator and a regular user. The moderator is a team leader, a project manager, a Scrum Master or a Product Owner, who manages and controls the ongoing activity itself. The moderator is required to have his/her own account created, be logged in, and all individual activities and events will be established under this account. Another user role, as I mentioned above, is a regular user. In most cases a regular user is a developer. It is not required for this user role to be registered and to have an account in the application. A regular user will be able to connect to the already existing event, by getting an invitation link. The home page of the “CPS Agile” application will contain basic information about the application and events to choose from to create a desired activity – Retrospective, Brainstorming or Poker Planning. While choosing and creating an activity the application will generate a unique link that the moderator will send to a team by regular communication tool, which is used on the case project. By clicking on the link, a regular user will be able to join the event after entering their name to the textbox. After completing all operations, which were described above, the moderator can start a session and regular users become a part of the event.

5.1.2 Retrospective

Each participant of the event has a task to come up with the ideas for improvement, which can be divided in three categories: Start, Stop, Continue. A participant enters and assigns their points separately from other participants, thereby no one can see ideas of others and it does not influence the proposals. Points are inserted using a simple form below

the selected category. If the participant is not satisfied with the text of his point, he can edit a text or delete it. As soon as all team members are ready to post their points or agreed time runs out, the moderator ends the first phase of the retrospective.

In the second phase, each member reveals a point by point. It is the important requirement, that points are revealed gradually, because otherwise individual members would be optically disturbed by reading all points at once. Each participant can reveal only his or her own point, not points of others. At the end of Retrospective event it is important for the application to be able to export the outputs, so the moderator and a team can keep it and work on it later.

5.1.3 Brainstorming

The main required policies and functionalities are:

- No criticism. Published ideas should not be able for comment by anyone. Even the worst and irrelevant ideas can inspire others to come up with a good solution.
- Write everything. The more ideas are written down, the more likely it is to find the best solution.

Each user puts his or her points separately from others and, when user is satisfied with their point, he or she can share it with others to be inspired. The user can overwrite or delete their points unless it has been already shared. At the end of a session it is essential to write and retain the outputs from Brainstorming, thereby the application should be capable to export the results of the event.

5.1.4 Poker Planning

During the Poker Planning session, a moderator writes down User Stories. Each User Story is explained and discussed with a Development Team at the beginning of Poker Planning event. Then each team member receives online cards, which contain numbers from Fibonacci sequence and one special card with a coffee cup symbol. The coffee cup represents a “coffee break” request. Planning meetings can take a long time, and this is a way for participants to request a small break. When everyone has a clear picture and knows what the User Story means, every participant separately selects a number that represents the workload of the current user story. If the selections match and all participants are agreed on similar workload, the User Story is estimated, and the selected number is written down. If great differences in estimations have occurred, a discussion on User Story starts and participants with the highest and with the lowest number advocate their choice. After the discussion all participants are voting again. If there is still no match, the highest number will be written down as an estimation, even though only one participant voted for it.

5.2 Discussion with the project managers and Scrum Masters

In this section of my paperwork I use information and knowledge, which were gained from the personal discussions with the project managers and Scrum Masters, who has experienced leading and managing distributed teams. Three out of four asked Scrum Masters had experience with a virtual team. Firstly, I am presenting asked questions and the most frequent answers on them. Secondly, I am reflecting on the discussions and making general conclusions, which will help me with the final list of the requirements on the web application, which supports agile techniques.

5.2.1 Asked questions and frequent answers

In this subsection I am providing a list of asked questions and the most frequent answers on them. Besides the questions on special features of agile techniques I as well consulted a design and graphical user interface of the desired application. The first question was:

- 1) What agile methods do you use and how often?

From the answers I found out that Stand-up meeting is carried daily or every other day by almost every team. Retrospective is used by the vast majority of the teams once or twice a month, it depends on a duration of the iterations and frequency of the releases. The Brainstorming was said to be used only at the beginning of the project and especially when developing a new product from the scratch. Poker Planning was used rarely, only on the specific projects.

The second question was put to see if there is a need for online tool among managers, who leads virtual teams. The second question sounded:

- 2) Would you likely to use a web application, which supports agile methodologies? If yes, which agile techniques would you like to see in it?

The result of the answers was affected by the size of the team. In small teams up to 3 – 5 people there were no need in a special online tool, inasmuch as regular communication tools were sufficient. But project managers, who lead the team of five and more people, all said they would definitely try to use it. The most desired feature was Retrospective ceremony. But for the daily Stand-up no one would use a special tool, because of its basic characteristics – high frequency and short duration.

The last question, which I was interested to get answer for, was about general functions, which were expected from the application by users.

- 3) What basic features should this application have?

Majority of the responses clearly pointed to a simple design, fast and no time-consuming login or account creation and last but not least intuitive user interface. Moreover, several Scrum Masters highlighted that there is no need in a creation of profile for each participant, only special moderator profile was taken as necessary.

5.2.2 Concluding propositions from the discussion

In this subsection I am summarizing comments, remarks and information which was gained from the discussion with the project managers. Along with the useful responses on the questions I as well got practical and valuable remarks regarding user expectations on the application. In regard to Retrospective feature it was purposed that each user will be limited in the number of possible votes. This specified number of votes will be defined by the moderator while setting up the event. These votes can be used to vote for the points of other users, but the user can give only one vote per each point. Another suggestion for the web application was the panel, where currently connected users can be seen. For better visibility it was suggested to assign a unique color for every user, thereby the points, which are shared with others, are in addition highlighted with color, which was assigned to the particular user. As the discussion with project managers showed up, an essential requirement on the application is the possibility to export data at least in CSV format. At the end we came to the conclusion that, although the application should be responsive, there is no particular need in the adapting of web application to the mobile devices, as all members of the team are using laptops on the meetings.

5.3 Usage of the web application

The proposed application is especially suitable for the middle size and large teams. It will be used and especially valued by the project managers and Scrum masters, who are managing the distributed teams. The main benefits are seen in the ability of the controlling the progress as a moderator and the reduction of the time, which are spend on the events, as there is no need to transfer the written points from sticky papers to an electronical version later.

The application is not only suitable for the software engineering industry, it can be used across different fields and disciplines. For example, Retrospective feature of this web application can be utilized while managing a long-term running project. It is beneficial to project success to ask and share an opinion of team members at least once or twice a month regardless of the field of the project. Also, the Brainstorming feature is a great tool for creating new ideas on the beginning of any project. And last but not least, the Poker Planning is an excellent technique for providing estimations.

5.4 List of the final requirements

Based on the analysis and consultations with the project managers, which were conducted in the previous subsections, the final list of the requirements was established and agreed with the client. The problem-solving area of the application is restricted to three basic agile methodologies - Poker Planning, Retrospective and Brainstorming. The main most important and weighted requirement from the client is separating users by two types – a moderator and a regular user. Only the moderator can manage the progress of the event and create new activity.

The final requirements on the application are divided into two groups of the functional and non-functional requirements. The functional requirements are defined both generally and then separately for each of three features – Retrospective, Brainstorming and Poker Planning. It was agreed to provide a prototype design of the web application, which can be later tested and evaluated by the members of the team.

5.4.1 General functional requirements

In this subsection I am listing the general functional requirements on the web application, which were approved and authorized by the client.

- GFR1 – In order to create the events, you must log in to the application as a moderator. In other words, only the moderator has rights to create and start the event.
- GFR2 – Regular users do not need to register and have an account in the application, they can join the created event only by writing down their name.
- GFR3 – While creating an event, the moderator fills the required parameters of the event and then a unique web address is generated and can be distributed to the team as a link to the event.
- GFR4 – A list of the connected users is shown to the participants. The list is updated in real-time, which means, that after a new user enters the event, it is immediately reflected in the list.
- GFR5 – Complete information and the results of the event can be downloaded as an Excel file at the end of the activity.
- GFR6 – Any user can register to the application in order to create their own account.
- GFR7 – In the application the user can login into his/her account and likewise logout.

5.4.2 Functional requirements on the Retrospective feature

This subsection is dedicated to the functional requirements of the Retrospective feature.

- RFR1 – While setting up the event, a moderator can assign a particular number of votes, which will be used by every participant.
- RFR2 – Retrospective activity consists of three phases – writing the points secretly from others, exposing the points and voting on the shared points.
- RFR3 – Only a moderator can manage, set and switch between the particular phases of the activity.
- RFR4 – Regular users are creating points by using a simple form.
- RFR5 – User's points are added and distributed between three columns, which are named "Start", "Stop", "Continue".

- RFR6 – Each participant can announce that he has finished writing down their points and the points are prepared for presentation.
- RFR7 – The points are only visible to the author until they are revealed to others in the second phase.
- RFR8 – A unique color will be assigned to each user; thus it will be easy to distinguish between authors of shared points.
- RFR9 – Points can be deleted unless they are shared.
- RFR10 – Points are shared gradually one by one.
- RFR11 – Each participant of the event can add only one vote to a particular point.

5.4.3 Functional requirements on the Poker Planning feature

The functional requirements on the Poker Planning feature were defined as follows:

- PPFR1 – Poker Planning activity should include three stages - writing a User Story, the estimation and the discussion phase.
- PPFR2 – Only a moderator can manage and switch between particular phases.
- PPFR3 – Only a moderator can add new User Stories.
- PPFR4 – A moderator can add comments to User Stories in any time.
- PPFR5 – All users will use special cards with numbers and one card with a symbol of coffee for the evaluation of User Stories.
- PPFR6 – Selected cards with the points are displayed at the same time during the discussion phase.
- PPFR7 – If someone selects a coffee card, a moderator can pause or interrupt the current phase of Poker Planning activity.
- PPFR8 – The moderator can terminate the ongoing User Story after the first voting or repeat the voting phase again.

5.4.4 Functional requirements on the Brainstorming feature

- BFR1 – When creating a Brainstorming session, a moderator can set the defined number of votes for every participant.
- BFR2 – The Brainstorming feature consists of the two phases, a presentation of the ideas and voting.
- BFR3 – A moderator controls all two phases of the Brainstorming activity.
- BFR4 – The ideas in the first phase are written in a simple form.

- BFR5 – Each participant can signal that he or she has already written all his/her points and is ready for the discussion.
- BFR6 – A unique color is assigned to each user, thereby the visibility of the shared ideas is increased.
- BFR7 – Points can be deleted until they are shared.
- BFR8 – During the discussion phase each participant can vote on the particular idea only once.

5.4.5 Non-Functional Requirements

- NFR1 – The web application is functional and accessible via such web browsers as the last two versions of Google Chrome, Firefox, Microsoft Edge and Safari.
- NFR2 – Sensitive data such as passwords for user accounts are not stored by the application as a plain text, but the web application will use a proper hash function to save a sensitive data in the database.
- NFR3 – The application runs in English language. A support of other languages is not required.

5.5 Summary

To summarize a work in this chapter, the significant effort to gathering the requirements had been made. The discussion with the primary users of web application and exchange of views with the professionals in project management industry had been done, in order to better infer, how the current processes can be improved by the targeted “CPS Agile” web application. In addition, it was appraised that the application could be used in other businesses besides IT/ICT field. After the mutual discussion, analysis and consideration the final list of the functional and non-functional requirements on the prototype was produced and approved by the client. The general requirements and requirements on individual activities, such as the Retrospective, the Poker Planning and the Brainstorming feature, were accepted and authorized by the client, thereby the design phase could come forth.

6 Design of the application prototype

This chapter is dedicated to the design, architecture and a proposal of the final look of web application. The whole design is based on the final list of the functional and non-functional requirements, which were outlined in the previous chapter.

Firstly, I am creating and describing use case diagrams that define the roles of individual participants and their activities. Secondly, I am providing an entity relationship diagram (ERD), where I am focused on the physical ERD as it reflects the architecture of the database the best. The result of the entity relationship diagram is a database model, which provides a representation, how the data will be stored, managed and processed. Then I am commenting on the architecture of the prototype, particular layers of the application and the communication between them. At the end of this chapter I am presenting the user interface design, layout of the html pages and draft of the whole prototype.

6.1 Use Cases

The goal of this section is to design individual use cases regarding each required feature of the web application – the Retrospective, the Poker Planning and the Brainstorming activities. The famous online community for developers and other IT professions SearchSoftwareQuality⁴ defines a Use Case as *“a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal”* (SearchSoftwareQuality, 2019). In other words, a use case is a set of activities, which are triggered by actors and which the system needs to perform, in order to achieve the desired outcome. In this context a "system" is the web application and the "actors" are participants, which operate under the defined roles within the application. So, in this section I define the activities that the users of the application can perform. I also demonstrate the relationships between the actors and the activities. Same as the foregoing section about requirements on the application, the Use Cases section is divided into four subsections – the General, Retrospective, Poker Planning and Brainstorming subsections, which provide use cases for the specific activity.

This section provides not only the separated use cases, but also the use case diagrams. A use case diagram is a behavior diagram in Unified Modeling Language (UML). Use case diagram models the functionality of a system using actors and use cases. Specific use case diagrams, which are related to a particular activity, can be found in Annex A: UML Use Case Diagrams. For the use case diagrams I used Unified Modelling Language Notation. To draw and create

⁴ More about online IT community, which was mentioned in the text above, can be found on their web page <https://searchsoftwarequality.techtarget.com/about>.

these UML diagrams I used an online free web application Creately⁵. I found the Creately tool easy to use and practical to work with. The current work is saved on the real-time basis, thereby there is no chance to lose the progress due to, for instance, a bad internet connection. There is also a possibility to share a work with another users by email, thus it is undemanding and useful for cooperation, for example, in a teamwork. At the end of Use Cases section, I am providing a requirements traceability matrix to check that all current project requirements were covered by the outlined use cases.

6.1.1 Actors

The actors of the activities are defined by roles. In the prototype of required web application there are only two roles of users:

- A moderator, that controls the flow of the activity and invites regular users to join the activity.
- And a regular user, that receives a link from the moderator, uses the link to join the activity by filling his name out and participating in the current event.

6.1.2 General Use Cases

These general use cases are mainly related to user authentication and creation of the events. General Use Case 5 also describes an activity related to the export.

GUC1 – Registration

1. The use case starts with a user that wants to start using the application as a moderator.
2. The user chooses an option “Register”.
3. The application shows a web page with a registration form.
4. The user fills the registration form out and click a “Register” button.
5. The application shows a web page with the form to log in.
6. The user logs in by filling the form and after logging in the user can fully use the application as a moderator.

GUC2 – Logging in

1. The use case starts when a registered user wants to use the application as a moderator.

⁵ The tool can be found on <https://app.creately.com>.

2. The user chooses an option “Log in”.
3. The application shows a web page with a log in form.
4. The user fills the log in form and click a “Log in” button.
5. The application authenticates the existing user and after successful authentication the user can use the application as a moderator.

GUC3 – Logging out

1. The use case starts when the moderator wants to stop working with the application.
2. The moderator chooses an option "Log out".
3. The user is logged out and the application shows a login form.

GUC4 – Activity creation

1. The use case starts when the moderator wants to create an event for regular users to participate.
2. The moderator selects one of the available events on the main page and clicks button “Create”.
3. A window with the activity is displayed and the moderator fills out required information about the activity.
4. After a completion of required data, the user clicks “Create” button and the user becomes a moderator of the selected activity.
5. The name of the moderator is displayed in the panel on the left in a list of the connected users together with his role as a Moderator.

GUC5 – Export

1. The use case starts when a moderator wants to export information about the past events.
2. On the taskbar on the page the moderator clicks "My Events".
3. A table with the events, where the user had a moderator role, is displayed on the web page.
4. By clicking on the button “Export”, detailed information about the results of the event is downloaded as an Excel file.

6.1.3 Retrospective Use Cases

The use cases, which are related to Retrospective feature of the application, are described in this subsection. The UML scheme of the Retrospective use cases can be found as a diagram in Annex A. The user-driven activities are divided into four phases which are sequentially connected. The phases are comprehensibly presented on the UML use case diagram and are described as the Awaiting, the Preparing Points, the Presentation and the Voting phases.

RUC1 – Start the activity

1. In order to start the event, a moderator presses the “Start” button on the control panel and the activity starts with the phase of Writing the Points.
2. The button for adding a point is enabled to users.

RUC2 – Connect to the activity

1. This use case starts when a regular user wants to connect to the event.
2. The regular user connects to a web page by clicking on the link, which is sent by the moderator.
3. The regular user sees a form where he or she fills their name in.
4. By clicking on the "Enter" button, the regular user becomes a participant of the event.
5. When an event is in the Presentation or the Voting phase, the shared points are displayed to the user.
6. At the time when the user is connected to the event RUC3 is emerged.

RUC3 – A list of the connected users

1. A username of the connected participant appears in a unique color on the panel with the list of the currently connected users.

RUC4 – Create a point

1. The use case starts when an actor wants to create a point during the event.
2. In order to create a point, the actor presses the plus symbol under one of the "Start", "Stop" or "Continue" columns.

3. A modal window with the textbox is opened, thereby the user can fill the information about their point in.

RUC5 – Delete a point

1. When an actor wants to delete the specific point, he or she opens the window with the point and then clicks “Delete” button.
2. The specific point is removed and is not displayed anymore.
3. The actor can delete the point only if it was not yet shared with other participants of the event.

RUC6 – Switch to the Presentation phase

1. Only the moderator can see the "Switch to point presentation" button. By clicking on this button on the event control panel, the moderator transits the whole event to the Presentation phase.
2. After clicking on the "Switch to point presentation" button, the button “Share my points” is displayed to the users.

RUC7 – Share a particular point

1. An actor wants to share their point with other participants of the event.
2. The actor opens the window with the point and then he or she clicks on the “Share” button in the modal window.
3. The modal window is closed. The shared point is now appeared at the first place under the relevant column on the web page and the point is displayed in user’s color. The point can be shared only during the Presentation phase. After sharing the point is displayed to the moderator and other participants.

RUC8 – Ready to vote

1. An actor wants to show to other participants that he or she has written all their ideas and is ready to vote on the points.
2. The actor clicks on the "Ready to vote" button on the top panel.
3. In a list with the current connected users on the left panel a special character is emerged for this user's name.

RUC9 – Switch to the Voting phase

1. To switch to the Voting phase a moderator presses the “Start a vote” button on the panel.
2. All shared points can be now seen by the moderator and other participants of the Retrospective event.
3. The regular users can now see “Vote” button next to each shared point under the “Start”, “Stop” or “Continue” columns.

RUC10 – Make a vote

1. The use case starts in the Voting phase when an actor wants to vote for the displayed points under the “Start”, “Stop” or “Continue” columns.
2. The participant of the Retrospective event clicks “Vote” next the particular point to add their vote.
3. A number of votes next to that point is raised by one, and other participants see the actual number of votes next to each point.
4. The participant of the event cannot vote on a same point more than once.

RUC11 – Finish the activity

1. The use case starts when a moderator wants to end the event.
2. The moderator presses the “Finish Retrospective” button to complete the event.
3. All participants see the results of the event and can now download the results into Excel file by clicking on the “Export” button.

6.1.4 Poker Planning Use Cases

This subsection is dedicated to a description of the use cases, which are related to Poker Planning feature of the application. The Poker Planning UML use case diagrams is to be seen in Annex A. The Poker Planning includes three phases which are followed by each other. The phases are the Writing User Story, the Voting and the Discussion. All these three phases in context of the use case flow are shown on the Poker Planning UML use case diagram in the attachments’ section.

PPUC1 – Connect to the activity

1. This use case starts when a regular user wants to connect to the current activity.

2. The regular user connects to a web page by clicking on the link, which is sent by the moderator.
3. The regular user sees a form where he or she fills their name in.
4. By clicking on the “Enter” button, the regular user becomes a participant of the event.
5. PPUC2 is emerged, when the user is connected to the event.

PPUC2 – A shown list of the connected users

1. A username of the connected participant appears in the panel with a list of actual participants of the Poker Planning event.

PPUC3 – Write a user story

1. A moderator wants to create a user story.
2. The moderator uses the “Add User Story” button to open a new modal window, where he or she can enter details of the user story.
3. The moderator writes the required text into opened “New User Story” window.

PPUC4 – Switch to the Voting phase

1. A moderator wants to switch to the next phase - “Voting” phase.
2. The moderator presses the “Vote and Discuss” button on the control panel.
3. The regular users can now see the description of the User Story and cards with the different value for the voting.

PPUC5 – Pick a card

1. This use case starts when a regular user wants to rate a User Story, which is currently being discussed.
2. A regular user picks one of the cards, which are available to them.
3. A card with the value is assigned to the User Story.

PPUC6 – Coffee card

1. If someone from the participants picks a card with a coffee symbol, a window with the option to pause the Poker Planning event appears and it is shown to a moderator.
2. When a window with the option to pause the Poker Planning event appears, the moderator has an option to decline the suggestion of the interruption. In this case the event continues as it was before.
3. If the moderator agrees and clicks on a suggestion of the temporary stopping, the current event is paused.
4. When the moderator temporary stops the event, all regular users are seeing a window with information, that event has been paused for a coffee break.
5. The moderator has a right to unpause the event at any time, by clicking on the “Resume event” button.

PPUC7 – Make a comment

1. The use case begins when a moderator wants to comment on the presented User Story.
2. The moderator writes a text to a special textbox for comments below the User Story.
3. To send a comment and to show it to other participants the moderator confirms his or her actions by clicking on the “Send a comment” button.
4. The comment is now visible to all participants of the Poker Planning event.

PPUC8 – Switch to the Discussion phase

1. The use case starts when the moderator wants to end the Voting phase and switch to the Discussion phase.
2. The moderator presses the “Reveal the results” button.
3. All users can now see what cards have been selected by other participants.

PPUC9 – Start a second round on voting

1. If the values of the picked cards are dramatically different, and a moderator with other participants are feeling they need a second ballot, the moderator presses the “Start second round” button.

2. The event is transited back to the Voting phase, where regular users must again pick a card and assign it to the current User Story.

PPUC10 – Finish the activity

1. The use case starts when a moderator wants to finish the Poker Planning event.
2. The moderator presses the “Finish Poker Planning” button to end the event.
3. All participants can now click on the “Export results” button on the main panel, in order to download an Excel file with the final estimations of user stories.

6.1.5 Brainstorming Use Cases

Brainstorming Use Cases subsection is associated with and devoted to the description of the user-driven activities and showing a connection between them. The Brainstorming activities, which can be performed by actors, are divided into three phases – the Awaiting, the Presentation and the Voting phases. The relationships between the activities and the phases with the regard of the different actors can be seen in Annex A.

BUC1 – Connect to the activity

1. This use case starts when a regular user wants to connect to the current activity.
2. The regular user connects to a web page by clicking on the link, which is sent by the moderator.
3. The regular user sees a form where he or she fills their name in.
4. By clicking on the “Enter the event” button, the regular user becomes a participant of the event.
5. BUC2 is emerged, when the user is connected to the event.

BUC2 – A shown list of the connected users

1. A username of the connected participant appears in a unique color on the panel with the list of the actual participants of the Brainstorming event.

BUC3 – Start the activity

1. When a moderator is ready to start the Brainstorming event with the connected users, he or she presses the “Start” button on the control panel and the activity transferred to the Presentation phase, where participants of the event can start to write their ideas.

2. All participants of the Brainstorming event can now see the button “Write an idea” to press.

BUC4 – Create a point

1. This use case starts when a participant wants to add their idea.
2. The participant clicks on the “Write an idea” button.
3. A modal window with a textbox appears on the screen, where the participant can write their ideas down.

BUC5 – Delete a point

1. If a participant wants to delete their idea, he or she clicks on that idea and a modal window with that idea is opened.
2. In the modal window the participant clicks on the “Delete the idea” button.
3. The idea is deleted from the event and is not displayed anymore.
4. The participant can delete the point only if it was not yet shared with other participants of the Brainstorming event.

BUC6 – Share a point

1. An actor wants to share the written idea with other participants.
2. The actor presses the “Share the idea” button.
3. The idea is now displayed to others in a round shape figure on the main web page of the Brainstorming event.

BUC7 – Ready to vote

1. A user wants to show to a moderator and other regular users that he or she has done with their ideas and is now ready to continue with a next phase.
2. The user clicks on the “Ready to vote” button on the control panel.
3. A special character is displayed next to the user's name in a list with the currently connected users.

BUC8 – Switch to the Voting phase

1. In order to transit from the Presentation to the Voting phase, a moderator presses the “Switch to Voting phase” button on the control panel.
2. The participants of the Brainstorming event can now see a “Vote” button next to every shared idea.

BUC9 – Make a vote

1. The use case starts when a participant of the event wants to give their votes to the ideas, which are presently shown on the web page.
2. The participant clicks on the “Vote” button, which is next to the idea, he wants to evaluate.
3. Next to evaluated idea a vote number is increased by one.
4. Other participants see changes in a real time, thereby the actual number of the votes is always displayed next to each idea.
5. The participant of the event cannot vote on a same idea more than once.

BUC10 – Finish the activity

1. The use case starts when a moderator wants to finish the Brainstorming event.
2. The moderator presses the “Finish Brainstorming” button on the control panel, in order to finish the event.
3. All participants can now see an option menu to go to the main page or to see the details of the past Brainstorming event.

6.1.6 Requirements traceability matrix

In this subsection I am providing a requirements traceability matrix for each feature – the Retrospective, the Poker Planning and the Brainstorming feature. I also provide the traceability matrix for general requirements, which are corresponding with the general use cases. The traceability matrixes assist to check and to ensure that all system requirements are met and covered by the use cases, which are defined in the subsections above.

A requirements traceability matrix, which comes usually in the form of a table, is a regular practice in a software development process. The traceability matrix is used to help to estimate the completeness of a correlation between two baselined documents using

a many-to-many relationship comparison (Excentia.es, 2019). In my case the baselined documents are the designed use cases and the established requirements on the system.

The provided requirements traceability matrixes help to verify, that the high-level design is matched with the detailed requirements from the client. The traceability matrixes below verify that every functional requirement is covered by at least one use case. They also verify that there is no single use case, which does not relate to the system specification, which was defined at the beginning by the client.

For my usage of the traceability matrixes, I took every single requirement and placed it in the left column. Then every use case was placed across the top row. If a requirement in the left column is related to a use case across the top, a mark is placed in the intersecting cell. A blanked cell means no relationship between this requirement and use case exists.

Firstly, I want to present the traceability matrix, which investigates the coverage of general functional requirements. This traceability matrix is shown in Figure 13 Traceability matrix for the general requirements.

	GUC4	RUC2	RUC2	RUC3	GUC5	GCU1	GUC2	GUC3
GFR1	X							
GFR2			X					
GFR3	X	X						
GFR4				X				
GFR5					X			
GFR6						X		
GFR7							X	X

Figure 13 Traceability matrix for the general requirements

Due to the provided traceability matrix for general requirements, it is verified and confirmed that all general requirements are met in the high-level design of the prototype.

Next, I want to be sure, that all functional requirements on the Retrospective feature of the web application were covered by the use cases, which were designed and presented in the subsections earlier. This specific traceability matrix can be found in Figure 14 Traceability matrix for Retrospective requirements.

	GUC4	RUC1	RUC6	RUC9	RUC11	RUC4	RUC8	RUC7	RUC3	RUC5	RUC10
RFR1	X										
RFR2		X	X	X							
RFR3		X	X	X	X						
RFR4						X					
RFR5						X					
RFR6							X				
RFR7								X			
RFR8									X		
RFR9										X	
RFR10								X			
RFR11											X

Figure 14 Traceability matrix for Retrospective requirements

With the help of the traceability matrix for the Retrospective requirements, it is proved and endorsed that all functional requirements on the Retrospective feature of web application are met.

Now I would like to investigate the coverage of the Poker Planning requirements. The results in a form of the traceability matrix can be seen in Figure 15.

	PPUC3	PPUC4	PPUC8	PPUC7	PPUC5	PPUC6	PPUC9	PPUC10
PPFR1	X	X	X					
PPFR2		X	X					
PPFR3	X							
PPFR4				X				
PPFR5					X	X		
PPFR6			X					
PPFR7						X		
PPFR8							X	X

Figure 15 Traceability matrix for Poker Planning requirements

The logical conclusion from the Figure 15 is that all client's requirements on the Poker Planning feature of the system were met and addressed with the designed Poker Planning use cases.

And finally, I want to put forward a traceability matrix for the Brainstorming system requirements. A corresponding matrix can be found below in Figure 16.

	GU4	BUC3	BUC8	BUC10	BUC4	BUC7	BUC2	BUC1	BUC5	BUC6	BUC9
BFR1	X										
BFR2		X	X								
BFR3		X	X	X							
BFR4					X						
BFR5						X					
BFR6							X	X			
BFR7									X	X	
BFR8											X

Figure 16 Traceability matrix for Brainstorming requirements

As it can be seen from Figure 16, the full coverage of all client's requirements on the Brainstorming activity was proved and verified against the presented use cases.

The obtained results from each of four traceability matrixes can serve as a proof of the successful design of the use cases. In addition, the designed interactions between the actors and the system, which are clearly defined in the event steps in each use case, can be used as the test cases later during the acceptance testing process.

As a next step, I am formulating and describing in the details a database model, in a form of physical ERD, which will be used while the implementation of the applications' prototype.

6.2 Entity relationship diagram

The aim of the subsection is to design a database model for the prototype of CPS Agile application. In this subsection I am suggesting a database model, which provides a representation of the database, how the data will be stored, managed and processed. As a form of the database model I chose to use an entity relationship diagram (ERD). An entity relationship model is a type of structural diagram, which is used in database design (Visual-paradigm.com, 2019). ER diagram visualizes entities and inter-relationships among these entities. Entities are defined as tables, which hold data that have their attributes. Relationships are the associations or interactions between entities. By defining entities, attributes and relationships between entities ERD illustrates the logical structure of database. One of the major characteristics and advantages of the ER diagram is the usage of the primary and foreign keys.

In my paper I decided to create a final physical ER diagram over conceptual and logical forms. Conceptual ERD is the simplest model among all three models and by itself is not

satisfying for the database design. Logical ER diagram represents information, which is gathered from the business requirements. Physical ER diagram models the actual design of the database, which is more relevant to my paper. In other words, the physical ER diagram represents the actual design blueprint of a relational database. It represents how information should be structured and how data should be related in a determined database management system (DBMS). To create “CPS Agile” schema, I used Enterprise Architect (EA) software from Sparx Systems. Enterprise Architect is a licensed tool, which helps businesses and organizations to model business processes and the architecture of a system as well as to implement these models through the whole life cycle of the development (Sparxsystems.com, 2019). EA is a very powerful tool for the architecture of enterprise systems, but it is sophisticated as well. The main disadvantage of this tool for me was that it is overcomplicated and confusing to use without a previous training. For this reason, I used EA in my thesis only for creation of a physical ERD schema.

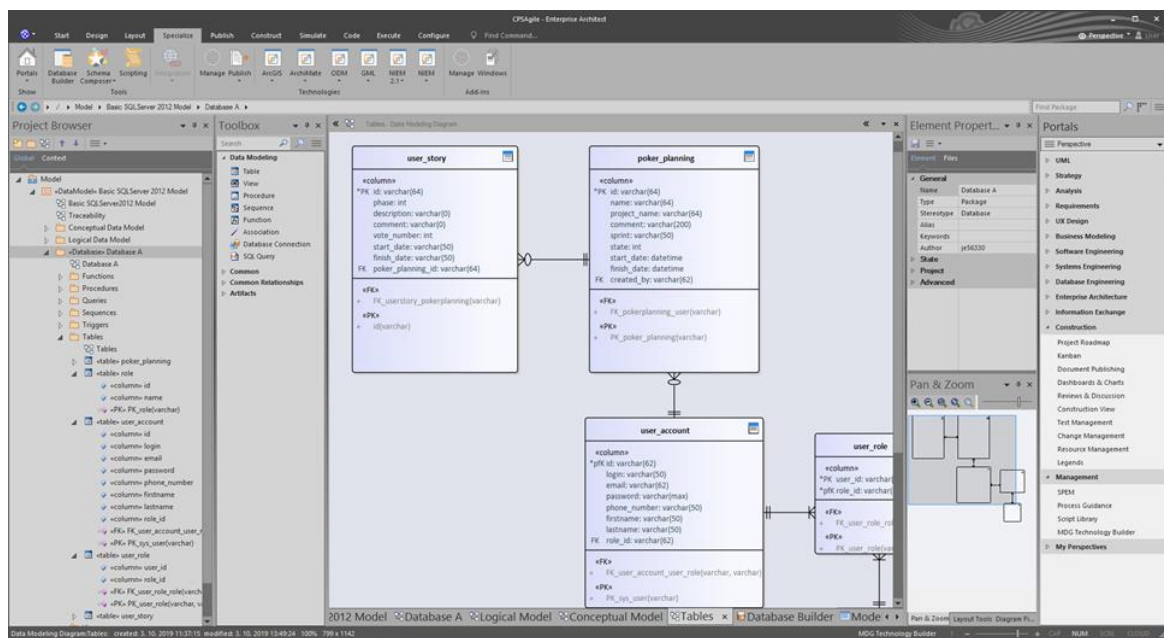


Figure 17 Process of creating ERD in Enterprise Architect software

As a result, the schema level design “CPS Agile” can be found in Annex B. In the phase of the implementation “CPS Agile” ER diagram is transformed into relational database. As a primary key of each table, a unique identifier of a row in a table, I used Globally Unique Identifier or GUID. Globally Unique Identifier is 128-bit unique reference numbers used in computing, thereby they are unlikely to repeat (Guid.one, 2019). This approach, with the negligible probability that two records will have the same identifier, allows the system to generate a primary key before saving a record to the database. Each available activity, the Retrospective, the Poker Planning and the Brainstorming, contains similar information about start and finish time, project name, a user, who created the activity and so on. Therefore, I decided to have one “event” table with “event_type” attribute, which determines what type of the activity is the event. The Retrospective points written by

users, same as User Stories written by a moderator and Brainstorming points, are stored in separate tables with the specific attributes. Associations of the “event” table depend on the activity type, which is defined in “event_type” table. It can have associations either with “user_story” or “brainstorming_item” or “retrospective_item” tables. Next, there are tables, which are connected to a users’ authentication. The “user_account” table contains information about registered users of the application. Sensitive information, such as password, is stored as a hash. The “role” table contains the roles, which are defined by the system. In case of many-to-many relationship, it is standard to include additional tables, such as a linked table. In the “CPS Agile” case it is “user_role” table, which resolves M:N relationship between “user_account” and “role” tables (at one time a user can have few roles, as well as one role can be assigned to few users). Inasmuch as a rough model of the relational database is defined, it is a right time to define an architectural pattern, which is going to be used during the implementation of the system.

6.3 Application’s architecture

This subsection describes an architectural pattern, which the web application is tight to. It is also related to an issue how the communication between different application’s layers is taking place in the system. The “CPS Agile” application is a web application that supports real-time client-server communication. The application is designed using a three-tier architecture. This type of client-server architecture is based on existence of three layers – a presentation tier, an application tier and a data tier, which are loosely coupled to each other. The presentation layer is the front-end layer and it consists of a graphical user interface. It is responsible for displaying a content and information to an end user in an appropriate format. The data tier stores information. It comprises of the database and data access layer. The application layer handles a business logic that supports the application’s core functions. This tier is accountable for solutions of complex business problems and ensures that the business rules are processed correctly. It also serves as an intermediary between the other two tiers. In “CPS Agile” application the business logic layer consists of few parts:

- Controllers
- Services
- Hubs

Controllers provide communication between the presentation tier and the application tier. A controller receives an input from user using UI, then processes the user's data with the help of services and then displays the results back to the user. Services, or components with custom interfaces, provide an additional logic to the application, for instance, functions to manage the users, which are currently logged in. Services are also used to operate on the data sent to and from the data access layer. The last part is hubs. Hubs are responsible for client-server real-time communication. Hubs are run on a server side and route messages in and out to make sure they get to a recipient in a real time.

A three-tier architecture has many advantages and benefits. In this paragraph I would like to point out some of them. Firstly, a speed of development can be increased due to

separation of different layers. With three tiers, each tier can be developed concurrently by different team of developers using different technologies and programming languages. As a conclusion each tier can be implemented with greater speed and greater quality than developing a singular code base. Secondly, scalability of a code is dramatically improved. A three-tier architecture provides the ability to upgrade a specific layer with minimal impact on the other layers, thus it is easier to evolve and enhance a product. Separating business-related functions from database-related functions enables to scale each layer independently at any time as new needs arise, without making changes to other tiers. Improved maintainability and performance of the system are other great advantages of three-layer architecture.

For the better understanding of a three-tier architecture the illustrated scheme of this type of a client-server architecture is presented in Figure 18.

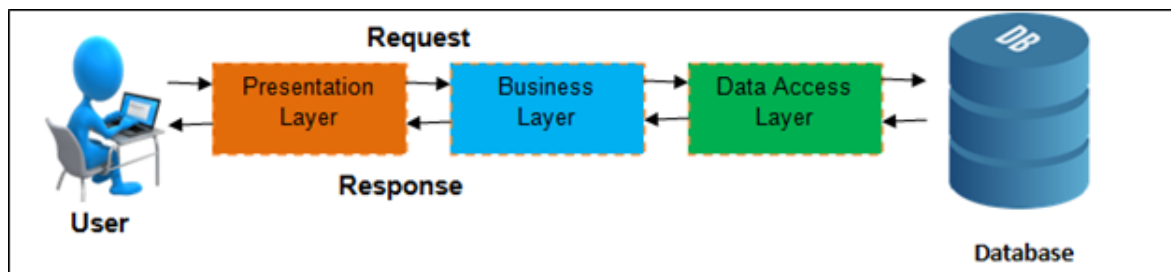


Figure 18 A three-tier architecture (Kaushik, 2016)

After the description of an architecture which is going to be used in the implementation of the prototype, the next step is to define how the web pages will look like to user. It means I am moving on determining graphical user interface and illustrating layouts of the web pages.

6.4 User Interface design

This subsection is dedicated to a draft of graphical user interface design. The GUI draft is presented by creating webpage wireframes for each feature of the “CPS Agile” application. A wireframe is a visual representation of a user interface, the page layout or arrangement of the website’s content, including interface elements, navigational systems, and their interaction together (Garrett, 2011). Website wireframes are often compared to the classic blueprints for any construction project, basically they represent the skeletal framework of a website (Brown, 2011). A wireframes building is an effective way to make rapid prototypes of pages and at the same time to measure functionality and practicality of a design concept in general. The key goals, which I want to reach by creating wireframes, are:

- ✓ To show the positioning of the content and visual elements of the design,
- ✓ To confirm and validate a layout of the “CPS Agile” web pages with the client.

The wireframes are constructed in basic colors, lacks typographic style or graphics, since the main focus is put on functionality and content priority. To create wireframes, I used MockFlow WireFramePro tool⁶. With ready-made components for any user interface such as web or mobile applications, it was easy and convenient to work with, including the possibility to export work in a wide range of formats – .png, .jpeg, MS Word, Adobe PDF, MS PowerPoint, HTML or Wire template. I have used a desktop version of the application, thereby I could work on it offline and without internet connection. The GUI drafts are created for each feature of the application – the Retrospective feature, the Brainstorming and the Poker Planning. The draft of the starting Home page of the “CPS Agile” application can be seen in Figure 19.

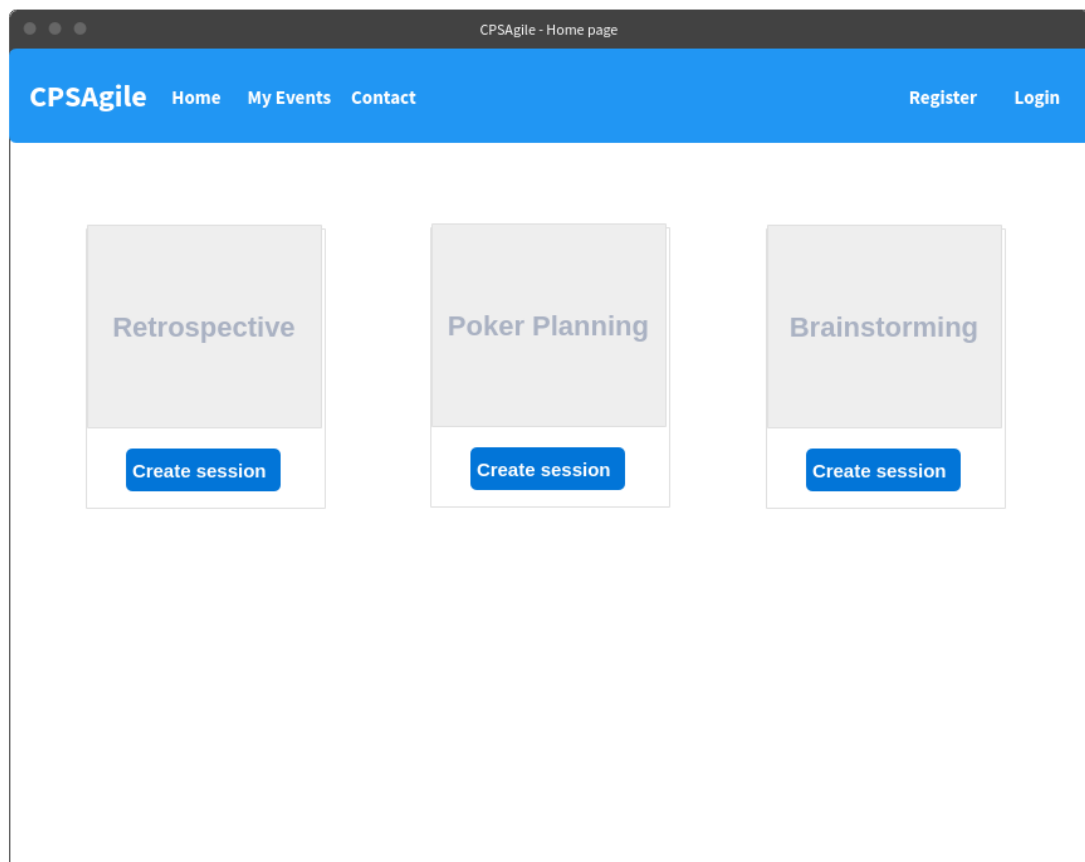


Figure 19 Home page of "CPS Agile" application

From Home page a user can register or login into existing account. After the login user can select which session he wants to start as a moderator. After the selection of the specific event, the window with the required information to fill in is appeared on the screen.

⁶ The tool is available on <https://mockflow.com/apps/wireframepro/>.

CPSAgile - Event creation

CPSAgile Home My Events Contact

scrum_master@mail.com Log out

Create event:

Project name:

Sprint:

Comment:

*Another fields,
depending on the event type*

Start

Figure 20 Page to create an event

Then a displayed web page layout is dependent on the concrete feature of the application and described further ahead in the corresponding subsection.

6.4.1 Retrospective event GUI

On the next Figures 21 and 22 there are webpage wireframes with basic components, which are used for the Retrospective feature. On the left panel a list of currently logged in users is displayed. A GUI draft for a regular user is to be seen in Figure 21. For a regular user the upper panel is filled with basic common information – a link on the current session to invite other team members, a time left till the discussion phase and “I am ready” button, which signals that the user has written all his or her points and is prepared to switch to the next phase.

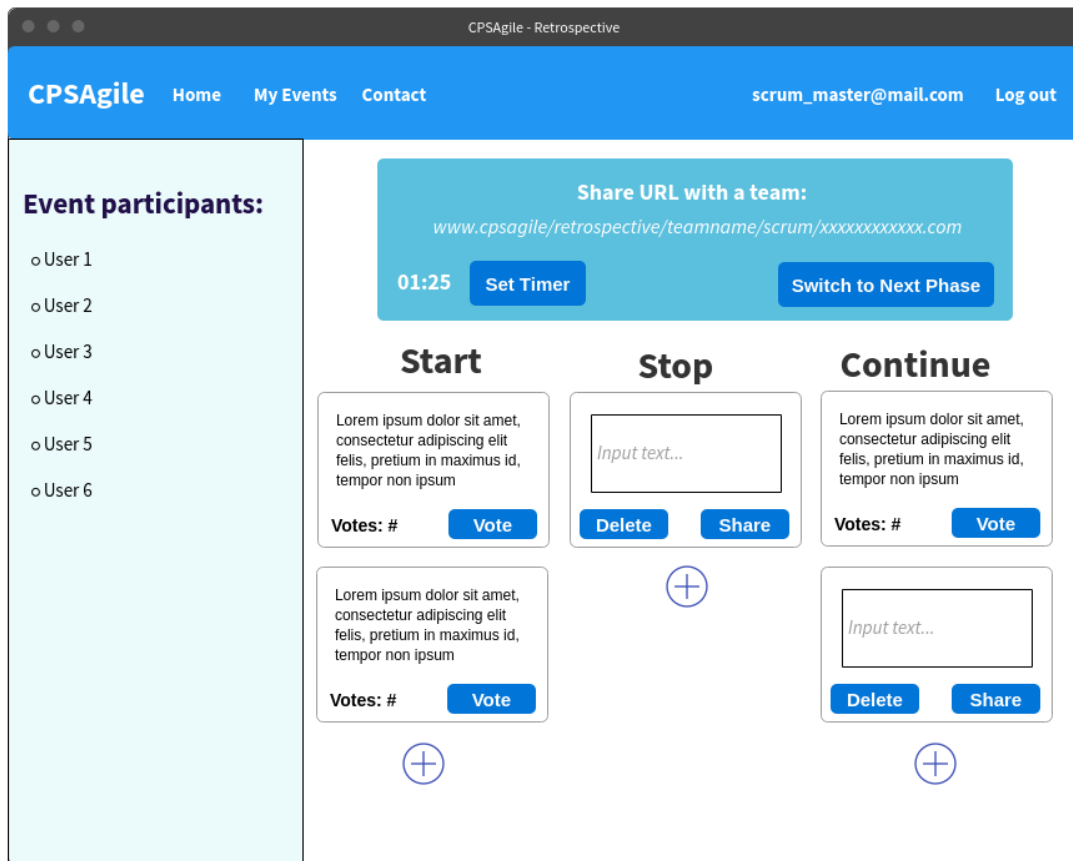


Figure 21 “Preparing points” phase from a regular user’s perspective

Figure 22, on the contrary, reflects webpage layout from a moderator’s perspective. On the upper panel the moderator, in addition to a sharing link, can set a timeframe for the current phase and control the course of the event by ability to switch to the next phase.

For both roles the main part of the page is the same. There are textboxes for retrospective points under three columns – “Start”, “Stop” and “Continue”. New points are added using the plus symbol below each column. Newly created points include “Share” and “Delete” buttons. Points, which have been already shared, are having a voting button and next to the button is an indicator with a current number of the submitted votes. The discussion phase later does not have much effect on the web page layouts, thereby it was left from the GUI drafts out with a preliminary agreement with the client.

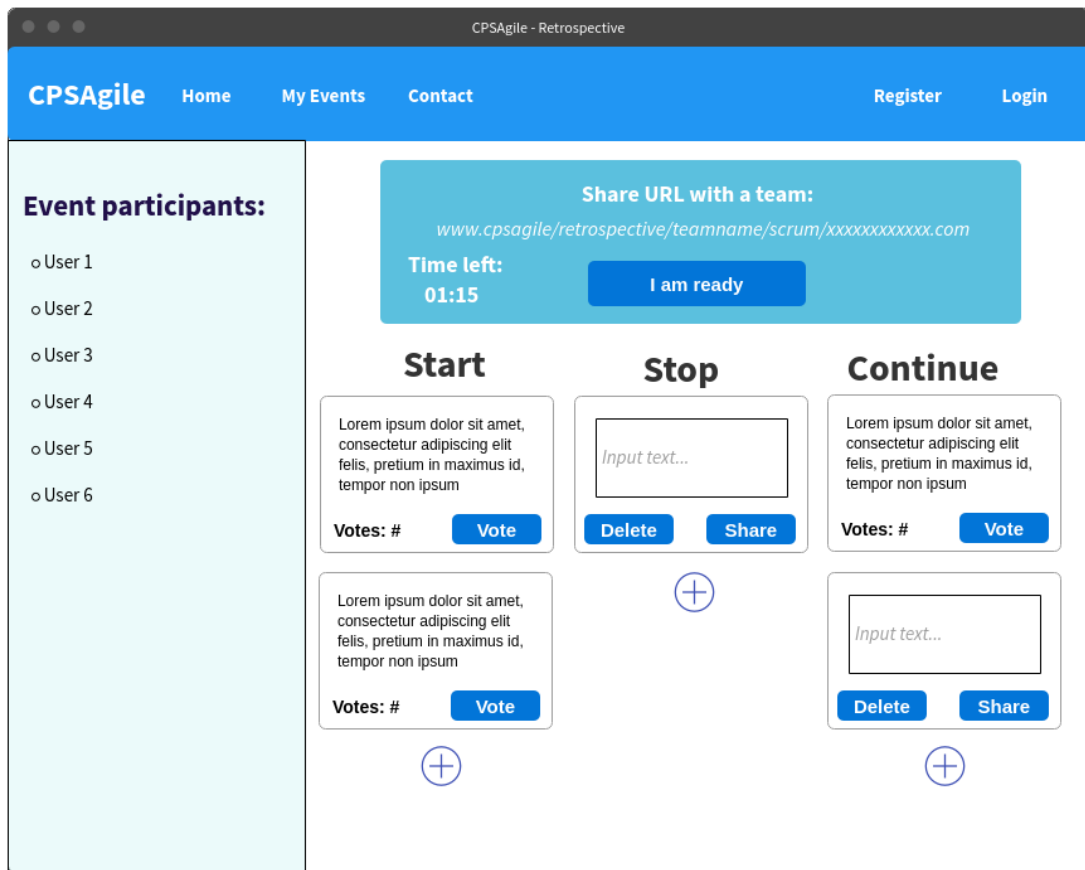


Figure 22 “Preparing points” phase from a moderator’s perspective

6.4.2 Poker Planning – creating User Story phase

This subsection concentrates on the first phase of the Poker Planning feature – creation of the User Story. When the moderator selects a Poker Planning option from the Home page of the “CPS Agile” application and fills required information out, a User Story creation phase begins. The User Story creation wireframe can be seen in Figure 23.

The moderator can write few stories down and choose, which User Story to start with. The left panel with the list of the currently connected users is available throughout the whole event. The top panel displays a link on a currently running event and buttons for the event control, which are not seen by regular participants.

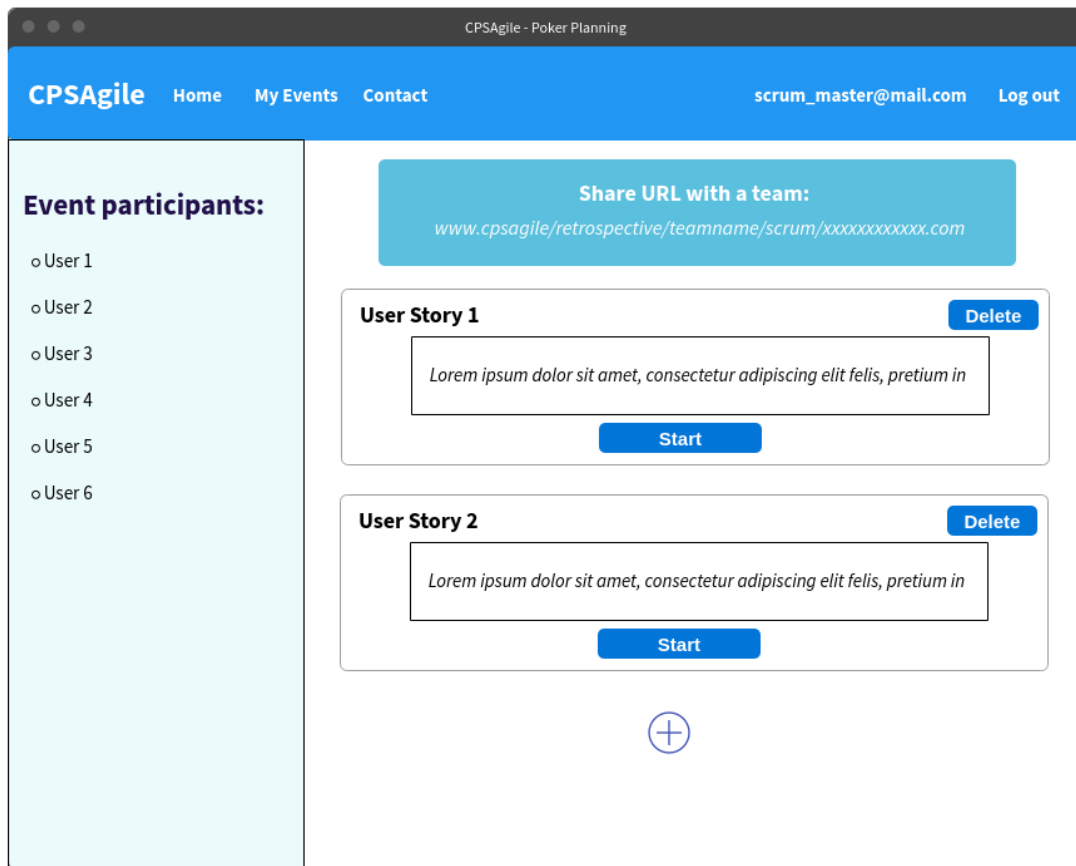


Figure 23 Poker Planning - User Story creation page, moderator's perspective

6.4.3 Poker Planning – voting phase

In Figure 24 and Figure 25 there are webpage wireframes with basic components, which are shown during the Voting phase. Under the top panel there are the cards with different values and the unique card with the coffee symbol on it. Under the cards there is a User Story to be voted on. The comments from a moderator are displayed at the bottom of the current web page. Figure 24 shows webpage layout from a moderator's perspective. The main differences from a regular user's view are the switcher to the next phase and a time setter on the top panel as well as the ability to add comments to the User Story during the whole event.

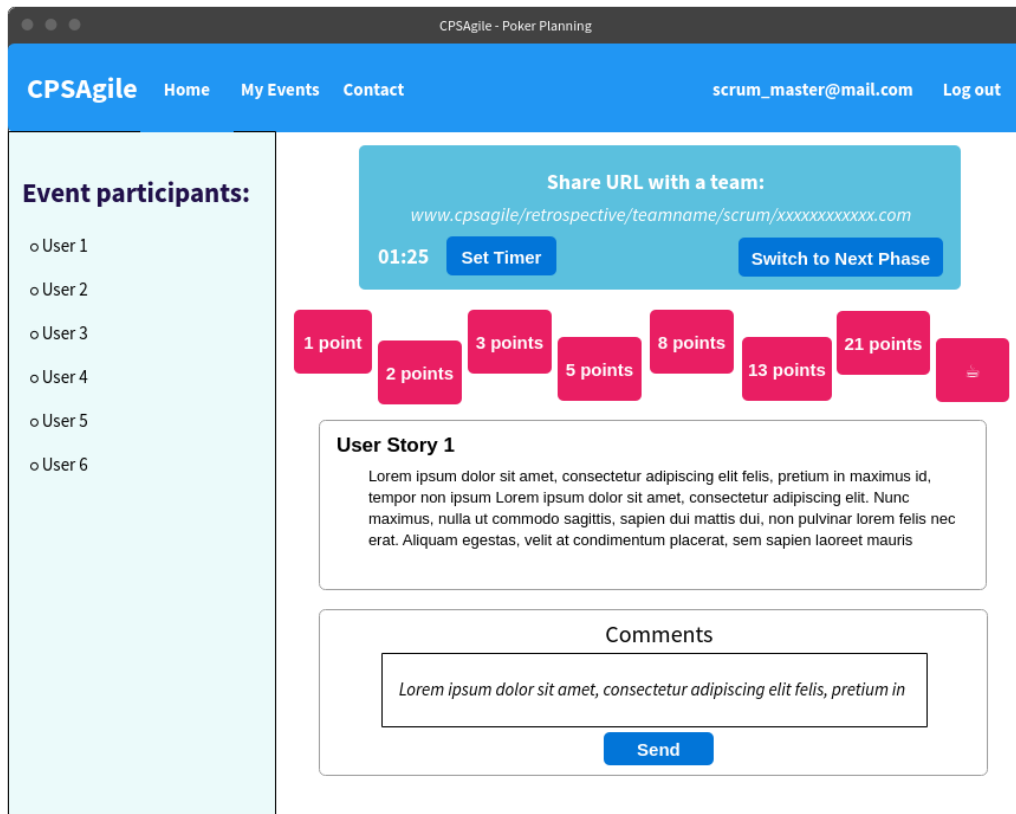


Figure 24 Poker Planning - voting phase, moderator's perspective

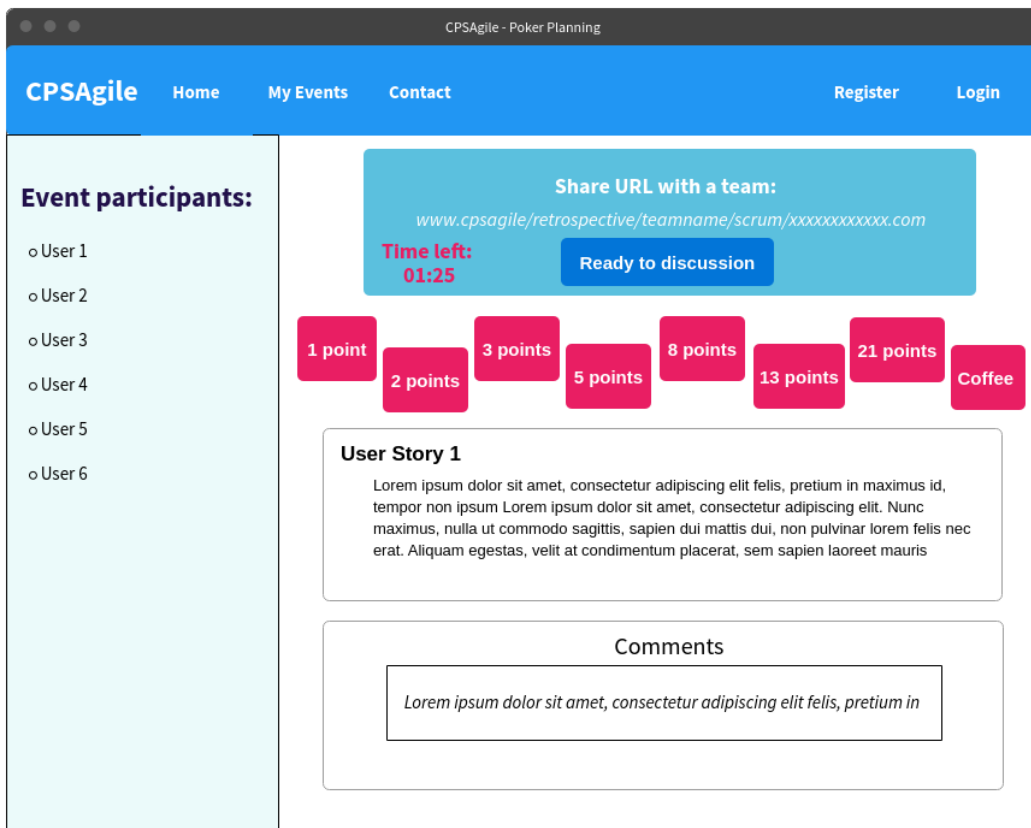


Figure 25 Poker Planning - voting phase, regular user's perspective

6.4.4 Poker Planning – discussion phase

This subsection reflects the web page layouts after the voting phase is finished. During the discussion phase every participant can see the results of the voting. A web page contains information about how many times each card was chosen and by whom. A moderator has two options to continue a session. The moderator can either finish current User Story and the event will be transferred to a User story creation page or decide to take another round of voting if the range of the selected cards were too contrasting.

The screenshot displays the 'CPSAgile - Poker Planning' web interface. The top navigation bar includes 'CPSAgile', 'Home', 'My Events', 'Contact', the user 'scrum_master@mail.com', and a 'Log out' link. On the left, under 'Event participants:', a list shows 'User 1' through 'User 6'. The main content area features a 'Share URL with a team:' section with a URL and two buttons: 'Start second round' and 'Finish User Story'. Below this, three horizontal bars represent voting results: '3 points' for '2x : User 4, User 5', '5 points' for '1x : User 2', and '8 points' for '3x : User 1, User 3'. A 'User Story 1' input field with a minus icon is positioned below the results. At the bottom, a 'Comments' section contains a text area with placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit felis, pretium in'.

Figure 26 Poker Planning - discussion phase, moderator's perspective

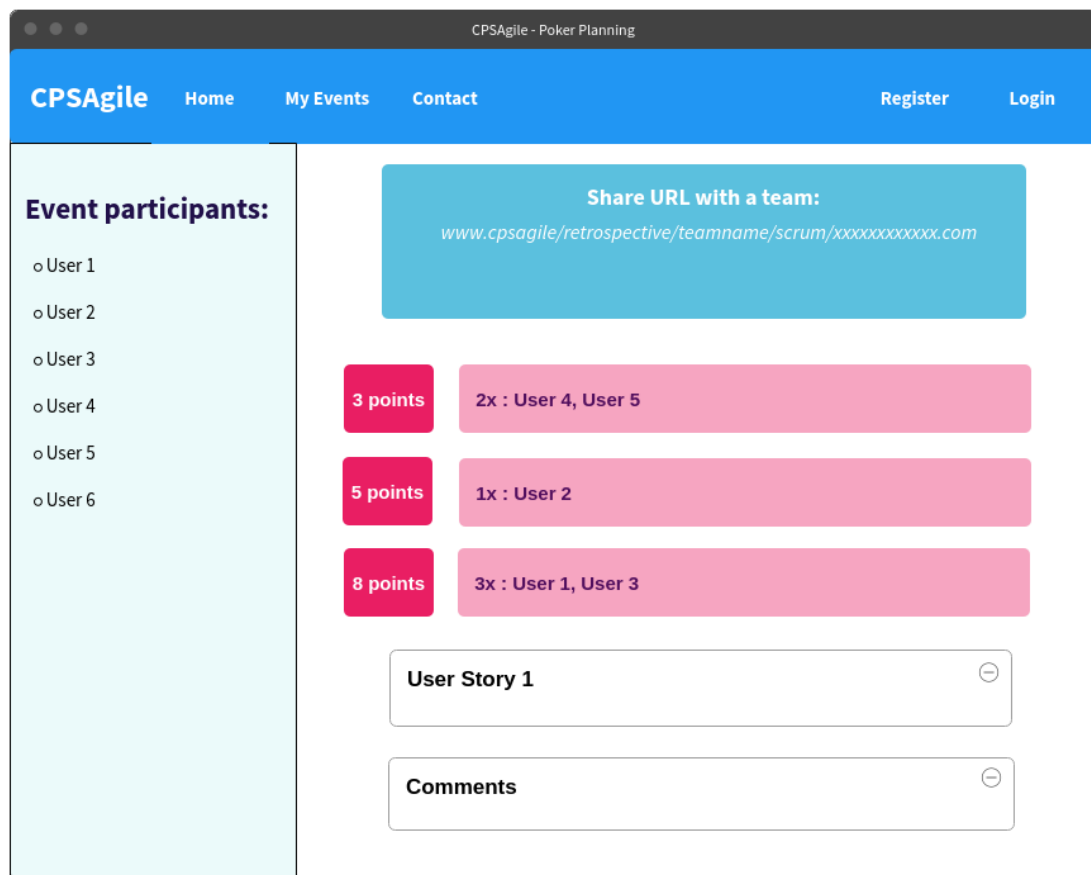


Figure 27 Poker Planning - discussion phase, regular user's perspective

6.4.5 Brainstorming event GUI

This subsection presents wireframes of the Brainstorming feature of the “CPS Agile” application. Similar to other features of the application, the left panel shows the currently connected users. For the moderator there is the possibility to set a timer and the button to transit the event to the next phase. In the page center there are points from the participants of the activity. The points’ components have the same design as the points from the Retrospective event with a single difference - the points are divided into two groups. The points, which are not yet shared, are displayed in the upper section of the page. The points, which have been submitted and are ready to be voted on, are shown underneath the first group.

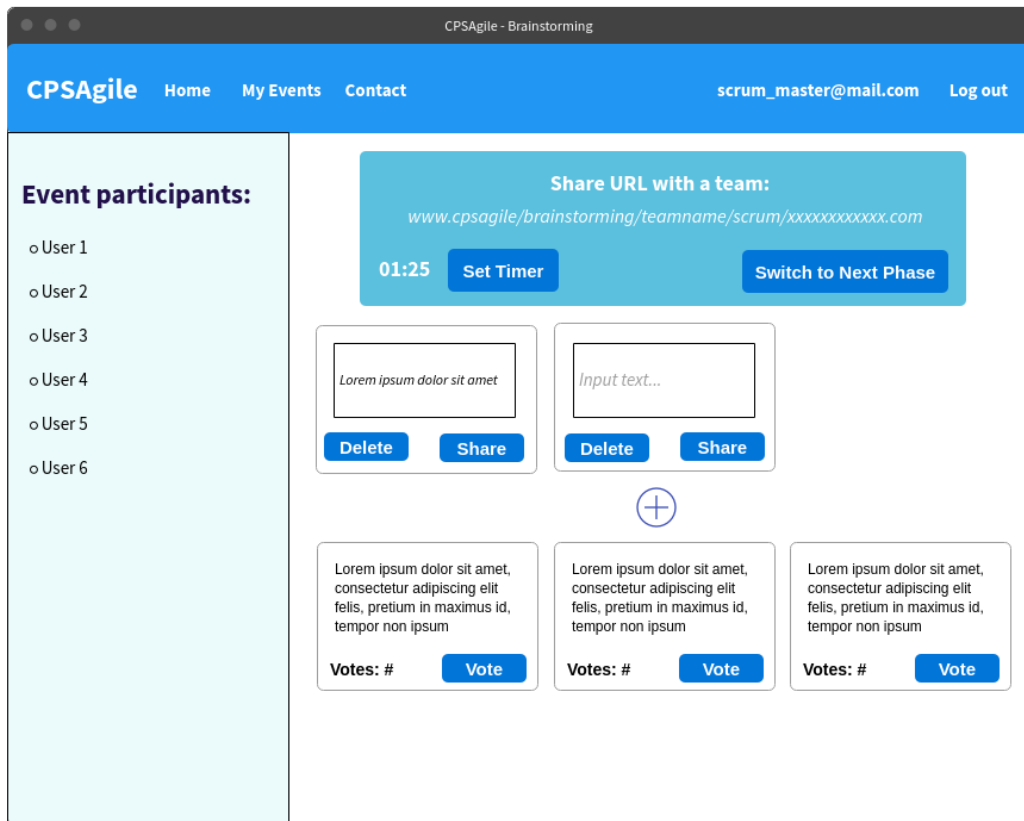


Figure 28 Brainstorming event from a moderator's perspective

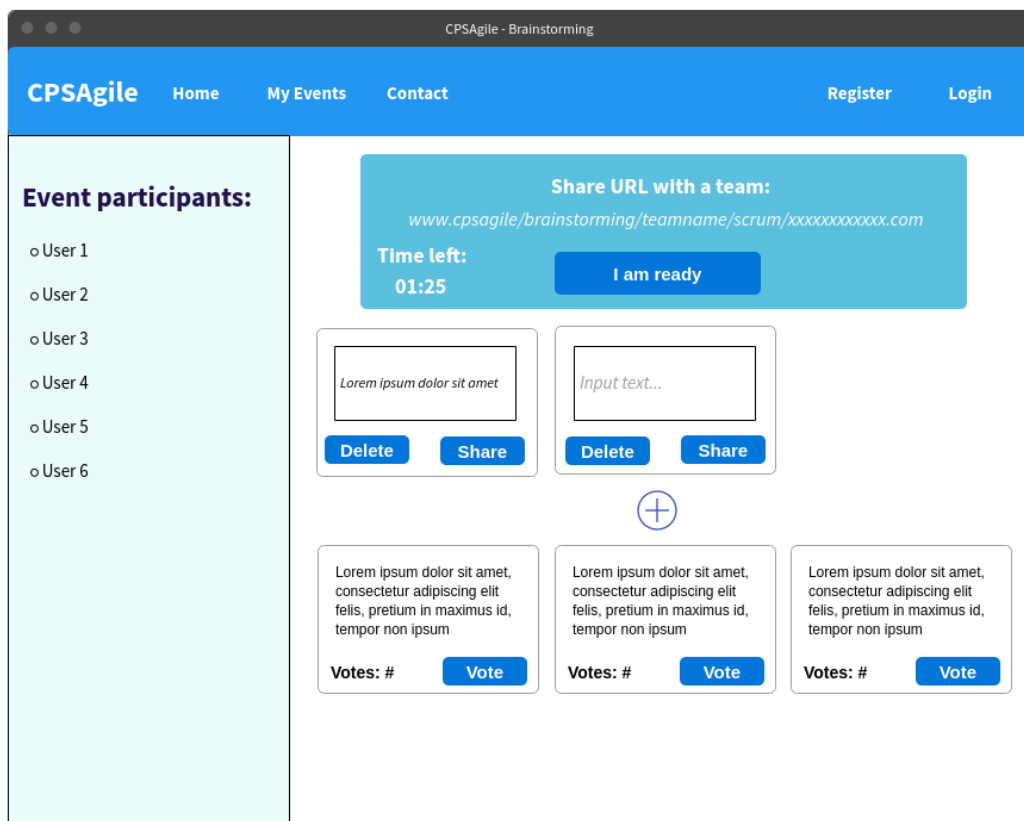


Figure 29 Brainstorming event from a regular user's perspective

6.5 Client's remarks and adjustments

The wireframes were presented to the client and after the discussion and minor remarks the final draft of the UI design was approved. The remarks concerned a timer feature and a color-setting for shared items.

The possibility of setting the remaining time till the next phase was reviewed by the client. The Scrum Master of the case project had decided that setting time is a redundant feature and can be removed, as he, as a moderator, manages the flow of the event and he decides when to switch to the next phase. The Scrum Master had also wished for a different color of already shared and not-yet-submitted points. When a point is shared by user and becomes visible for all connected users, it gains a unique color, which is the same as the color assigned to the user, who shared this point. The points, which have not yet been shared, are gray. Figure 30 shows the revised web page layout draft of the presentation phase.

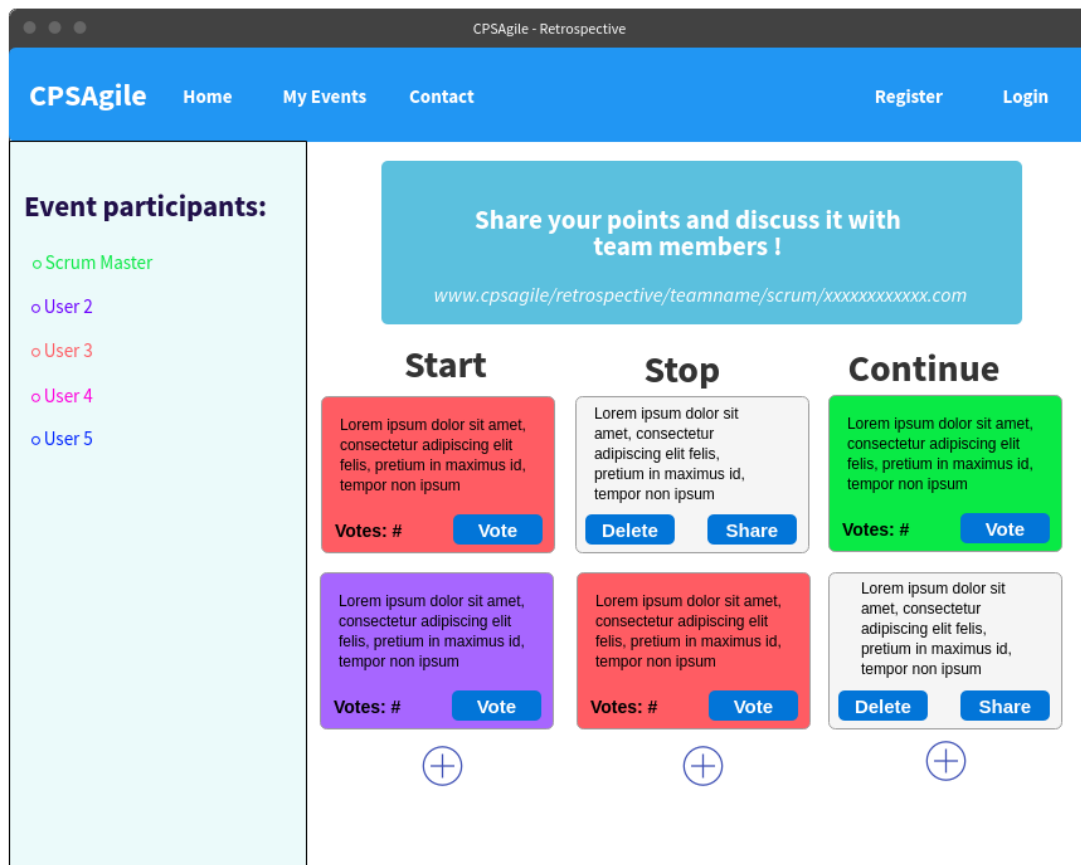


Figure 30 Final view of presentation phase after client's remarks

As Figure 30 indicates, all participants of the event can see, that a participant “User 2” shared the point, which is situated below in the “Start” column. The Scrum Master shared the point, which is located at the top of the “Continue” column. The “User 3” shared two points. The point at the bottom of the “Continue” column and the upper point in the “Stop” column have not yet been shared and therefore are not visible to other connected participants of the event.

6.6 Summary

In this chapter I described the main characteristics of the design of the “CPS Agile” web application. I established the use cases, based on the client’s requirements, and provided the use case diagrams. Then I double checked the coverage of the requirements with the help of traceability matrix. The results showed that all requirements, that are crucial to the customer, are met by the implementation of the use case scenarios. Furthermore, I designed the physical entity relationship diagram, which is going to be used while creating a database and its entities. It also represents, how the data will be stored, managed and processed. Besides the database model I described the internal architecture of the application as well. The “CPS Agile” web application is a three-layer client-server application. The data are updating at real-time, so all participants of the event see the actual state. At the end of the chapter the design of the user interface was presented and the consultation with the primary future users of the “CPS Agile” application was held. After the consultation the minor changes to the UI design were introduced and then the whole concept of the prototype of the web application was approved by the client. As the design of the prototype had been agreed and accepted by all parties, the implementation itself started.

7 Implementation

The chapter is devoted to description of implementation process of the prototype. Considering the scope of the Master thesis and time restrictions, the physical development, the demonstration and evaluation were reduced to Sprint Retrospective and Poker Planning features. The main aspects of the choice of the technologies to be used for the prototype development were a popularity in a developers' community and support of the object-oriented programming. The case project is running on the .NET platform and using C# programming language, therefore I chose to use C# language for the application prototype, so the team can later support and adapt the application according to their needs. To develop a prototype, I used ASP.NET MVC technology. The nature of Scrum ceremonies demanded a real-time communication between participants of the event, so I utilized an open source SignalR library to add real-time functionality to the "CPS Agile" application. The frontend part was implemented using JavaScript (JS), jQuery, HTML5 (Hypertext Markup Language) and CSS (Cascading Style Sheets) technologies. Entity Framework (EF) was used to communicate and execute SQL statements in a database. To manage the database I used Microsoft SQL Server, a relational database management system (RDMS). Visual Studio, an integrated development environment (IDE) from Microsoft, was used for developing the application. To control changes and manage the code was used version control system Git. The code was stored on collaboration and code hosting platform GitHub. The detailed description of used technologies and frameworks is provided further in this chapter.

7.1 Architecture

As it was described in Design chapter of my paperwork, the application is built on three-layer architecture. To reach this goal I utilized Model–View–Controller (MVC) design pattern, which helps to decouple user-interface (View), data (Model) and application logic (Controller). ASP.NET MVC framework provides a powerful, patterns-based way to build web applications using the MVC pattern that enables to achieve clean separation of main tiers. Client's requests are routed to a Controller which handles interactions with a Model, performs actions and retrieve data. Then the Controller passes information to a View. The View accepts information from the Controller and renders it to user interface. Interaction scheme within MVC pattern is presented in Figure 31. The Model represents the state of the application and any business logic or operations are performed by it. The View is using the Model that contains the data to display on the web page. The Controller creates and populates Model instances to the View. The View is accountable for presenting content on the web page. The View uses the Razor, template markup language, to embed programming code in HTML markup. Razor view engine is used to dynamically generate web content on the server; therefore, it enables to mix server code with a client-side code and content. I am using strongly typed Razor Views, that limits Controllers to pass only strongly typed Models to Views and enables the Views to have

IntelliSense⁷ and type checking. The View only displays information, while the Controller handles and responds to user interactions. The Controller is the initial entry point, it controls how the application responds to client's request. The Controller selects which Model to work with and which View to render.

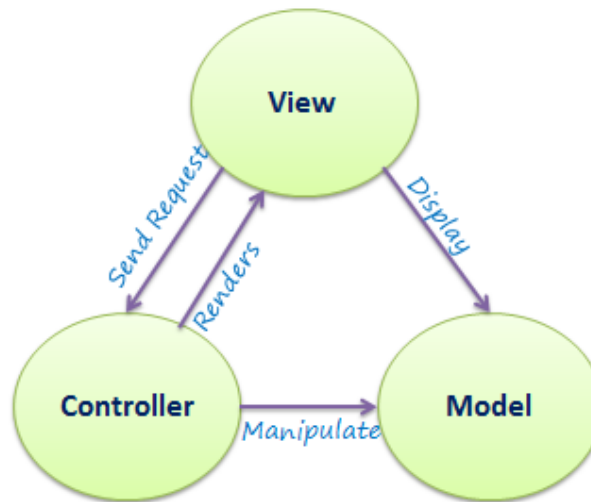


Figure 31 Interaction within MVC pattern

Both the Controller and the View is dependent on the Model, but the Model does not depend on other parts. This provides one of the key benefits of the separation – it allows to develop and test application logic independent of the visual content presentation.

The business logic layer (BLL) of the application is presented by Controllers, Services and Hubs. Services store and enforce all business rules. For example, File Manager service is responsible for exporting results of the Retrospective and Poker Planning events to Excel file. Following best practices, I implemented the application the way that each Service has its own application program interface (API). API describes how information is exchanged between different components of the application. All components of the application, mainly Controllers, are communicating with Services only through API. This approach ensures compliance with one of the five SOLID principles – Interface segregation principle (ISP). The ISP principle states that client should not be forced to depend on methods that they do not use, which can also mean that many client-specific interfaces are better than one general purpose interface. All application program interfaces of Services are stored in API folder. To ensure Dependency Inversion principle (DIP), I implemented Dependency Injection (DI) design pattern, which helps to decouple dependencies between layers through shared abstractions. Dependency Injection is one form of the Inversion of Control (IoC) principle. The relationship between IoC, DIP, DI and IoC Container are shown in Figure 32.

⁷ IntelliSense is an intelligent code completion tool with a variety of code editing features.

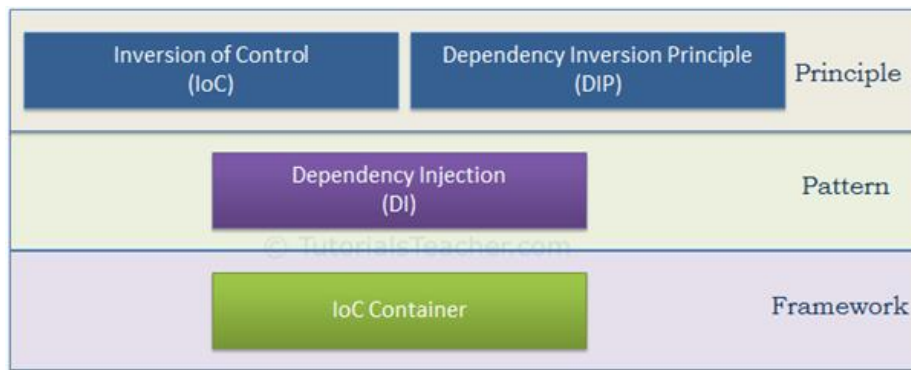


Figure 32 DIP, IoC, DI and IoC Container (Tutorialsteacher.com, 2019)

The IoC and DIP are high level design principles, they recommend certain best practices but do not provide specific details about the implementation. Dependency Injection is a pattern, which implements IoC principle to invert the creation of dependent objects. The IoC Container is a framework, which automatically injects dependencies throughout the application. IoC Container was not used in the implementation of “CPS Agile” web application. In Dependency Injection pattern there are three common ways, how to inject dependencies: through a constructor, through a property, or through a method. In “CPS Agile” application code I am injecting dependencies through a constructor. The example of DI can be found, for example, in FileManager class. The code is presented in Code 7.

Code 7 Implemented Dependency Injection pattern in FileManager class.

```

1 public class FileManager : IFFileManager
2 {
3     private IRetrospectiveService _retrospectiveService;
4     private IPokerPlanningService _pokerService;
5
6     public FileManager(IRetrospectiveService retrospectiveService,
7                       IPokerPlanningService pokerService)
8     {
9         _retrospectiveService = retrospectiveService;
10        _pokerService = pokerService;
11    }
12
13    public string ExportRetrospective(string retrospectiveId)
14    {
15        var retrospective = _retrospectiveService.GetRetrospective(retrospectiveId);
16        . . .
17    }
18
19    public string ExportPokerPlanning(string id) { . . . }
20 }

```


The third part of business logic layer is Hubs. Hubs are part of SignalR framework, and they are managing remote procedure calls (RPC) from a server to connected clients and from clients to the server (see Figure 33).

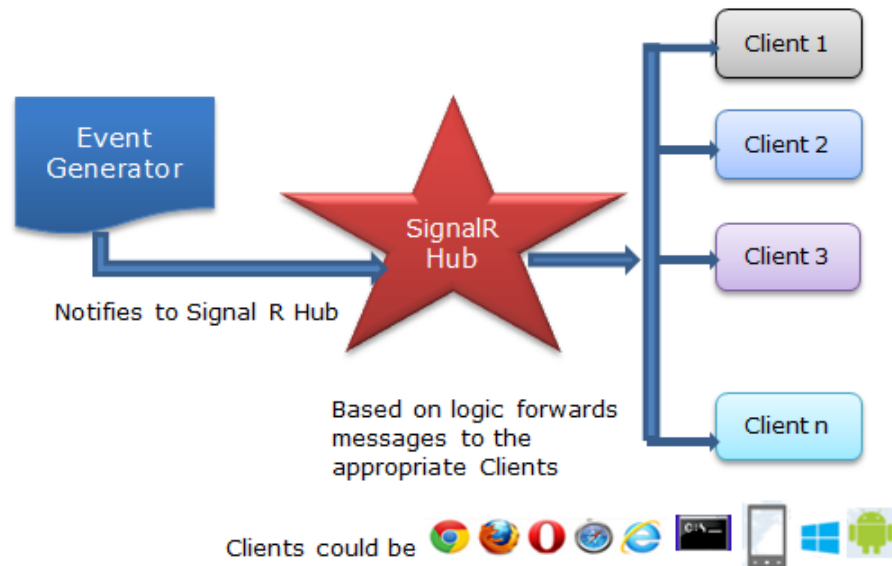


Figure 33 SignalR Hub communication (C-sharpcorner.com, 2019)

Methods that can be called by clients are defined in server code; from server code are called methods that run on the client. In client code, methods that can be called from the server are defined; and from client code the methods that run on the server are called. SignalR Hub is a primary point at which all communication between client and server occur. As the example of such communication can be Use Case, when Moderator adds comment to the running User Story. When Moderator wants to add or update comment to the current running estimation of User Story, he clicks on the button *btn_sendComment* and the client code (see Code 2) calls *addCommentToUserStory* server code in *PokerPlanningHub* (see Code 3).

Code 2 Client code, which calls *addCommentToUserStory* server code.

```

1 var pokerPlanningEventHub = $.connection.pokerPlanningHub;
2
3 $(document).on('click', '#btn_sendComment', function () {
4     var storyId = $("div[name='userStoryBody']").attr("id");
5     var commentOnStory = $('#userStoryComment').val();
6     pokerPlanningEventHub.server.addCommentToUserStory(storyId, commentOnStory);
7 });

```

In this example button *btn_sendComment* is an Event Generator, which calls server code in SignalR Hub. The server code processes the provided data and calls client code *updateComment* in all connected clients (see Code 3).

Code 3 Server code in Hub.

```
1 public void addCommentToUserStory(string storyId, string commentOnStory)
2 {
3     var connectionId = GetConnectionIdForUser();
4     var eventId = _userContainer.GetEventId(connectionId);
5
6     if (_pokerPlanningService.AddCommentToStory(storyId, commentOnStory))
7     {
8         List<string> receivers = new List<string>();
9         _userContainer.GetConnectionIdList(eventId, receivers);
10        Clients.Clients(receivers).updateComment(commentOnStory);
11    }
12 }
```

Server code *addCommentToUserStory* in *PokerPlanningHub* calls *updateComment* on all connected clients. Code 4 shows implementation of *updateComment* code on Moderator's client, Code 5 shows *updateComment* implementation on regular user's clients.

Code 4 Client code *updateComment* on Moderator's page.

```
1 pokerPlanningEventHub.client.updateComment = function(comment) {
2     $('#blink').fadeOut('slow', function() {
3         $(this).fadeIn('slow', function() {
4             $(this).fadeOut('slow');
5         });
6     });
7 }
```

Code 5 Client code *updateComment* on regular user's page.

```
1 pokerPlanningEventHub.client.updateComment = function(comment) {
2     $("#userStoryComment").empty();
3     $("#userStoryComment").append(comment);
4
5     blinkCommentUpdate();
6 }
```

The magic behind Hubs is responsibility of SignalR framework, which provides real-time communication between server and clients. The core principles of SignalR framework are described later in “Used libraries and frameworks” section.

The solution of “CPS Agile” application consists of only one C# “CpsBoostAgile” project, where all code is stored. The structure of the project with highlighted components of business logic layer can be found in Figure 34.

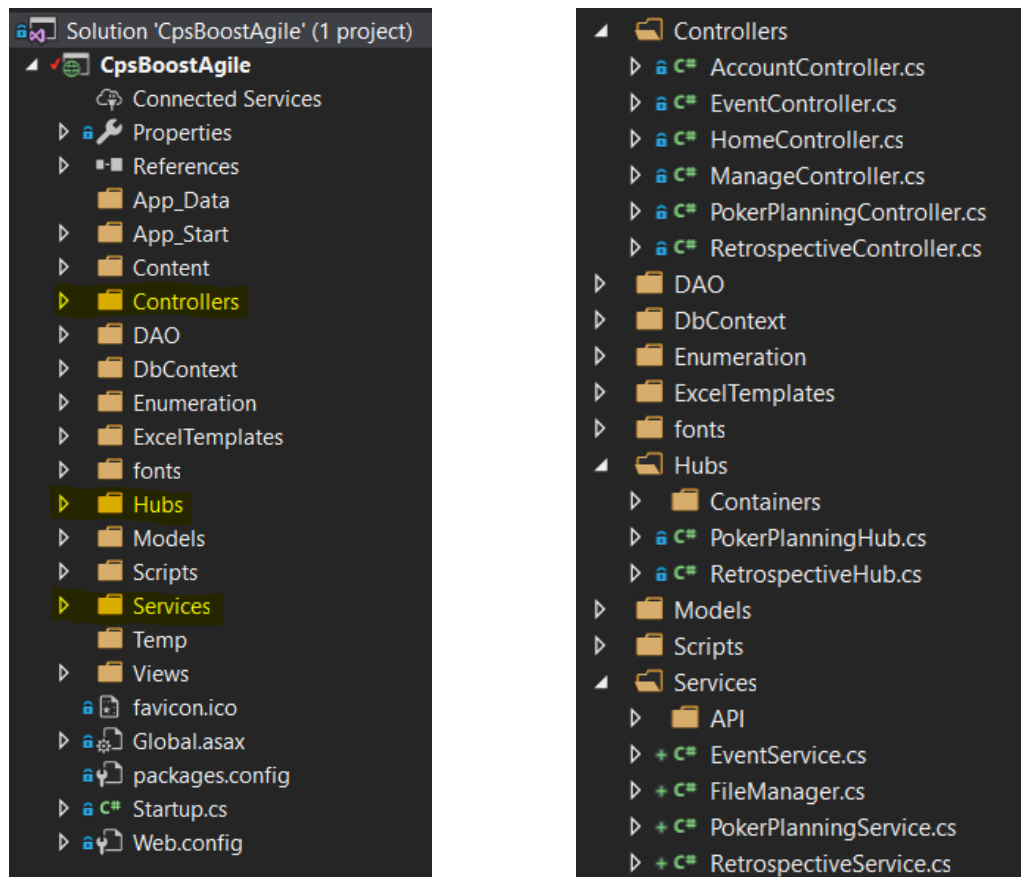


Figure 34 Structure of CpsBoostAgile project, business logic layer

The data access layer (DAL) is represented by data access objects (DAO) and database context classes. For communication with the database was used Entity Framework. EF is ORM framework that enables to work with the database using .NET objects and eliminates the need for most of the data-access code. DAO are simple objects that do not contain any business logic and provide data without exposing details of the database. The database context classes are primary classes that are responsible for interaction with the database and are used to persist data. It is a bridge between entity objects and the database. DbContext classes are also responsible for change tracking, caching, managing relationships and converting raw data from the database into entity classes (see Figure 35). The data access layer and application logic are connected through Services. The presentation layer and application logic are connected through Hubs and Controllers.

The presentation layer is represented by Views (.cshtml format files). The Views are using Razor syntax, scripts and CSS libraries, which are stored in Content and Scripts folders. CSS and JS libraries used in the application are listed in the next section of my paperwork.

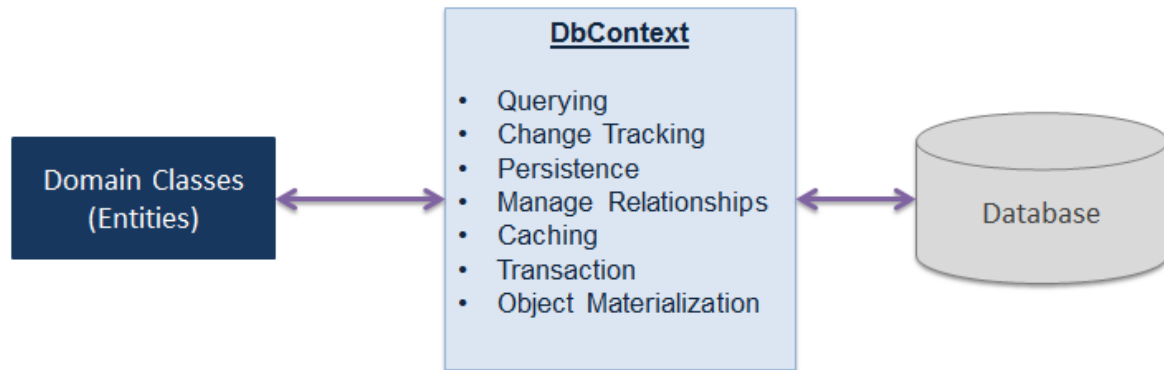


Figure 35 DbContext responsibilities in Entity Framework (Entityframeworktutorial.net, 2019)

7.2 Used libraries, frameworks and technologies

In this section I am describing the frameworks and libraries, which were used to implement the “CPS Agile” web application. The main frameworks and technologies, which were used for the implementation of the backend part, are C#, ASP.NET MVC v5.2.3, EF v6.1.3, EPPlus v4.5.3, SignalR v2.4.1, LINQ, ASP.NET Identity v2.2.1, OWIN v3.0.1, Newtonsoft.Json v6.0.4. For the frontend part of the application were primarily used HTML5, CSS, jQuery v1.10.2, AJAX, Bootstrap v3.0.0, Font Awesome v4.7.0, Clipboard.js v2.0.4.

7.2.1 ASP.NET MVC

The ASP.NET MVC is a web application framework, which extends .NET platform with libraries and tools for building web applications and implements Model-View-Controller pattern. It is an open-source framework, developed by Microsoft.

7.2.2 ASP.NET Identity and OWIN

The Open Web Interface for .NET (OWIN) is a standard interface between web servers and .NET web applications. It defines a specification how ASP.NET applications and hosting servers should be built without dependencies on each other. The OWIN and the ASP.NET Identity are integrated in ASP.NET MVC templates from Microsoft. The ASP.NET Identity is the membership system for authentication and authorization of the ASP.NET applications’ users. The ASP.NET Identity stores information about users in the database. It also allows to customize sign-in and login functionality, account lockout after the specific

number of invalid login attempts, supports two-factor authentication, role-based authorization and login via social channels like Facebook, Twitter and Google account.

7.2.3 SignalR

ASP.NET SignalR is a library that allows to add real-time functionality to web applications. It supports asynchronous notifications from server code, which are sent to the connected clients. A client is defined as any connected device, a laptop, a smartphone, a desktop computer, etc. The library includes server-side and client-side JavaScript components and provides API for server-to-client and vice versa RPC (calling JS functions in a client's browser from .NET code on the server side). SignalR handles connection management automatically and allows to send messages to specific clients or to all connected clients simultaneously (see Figure 36).

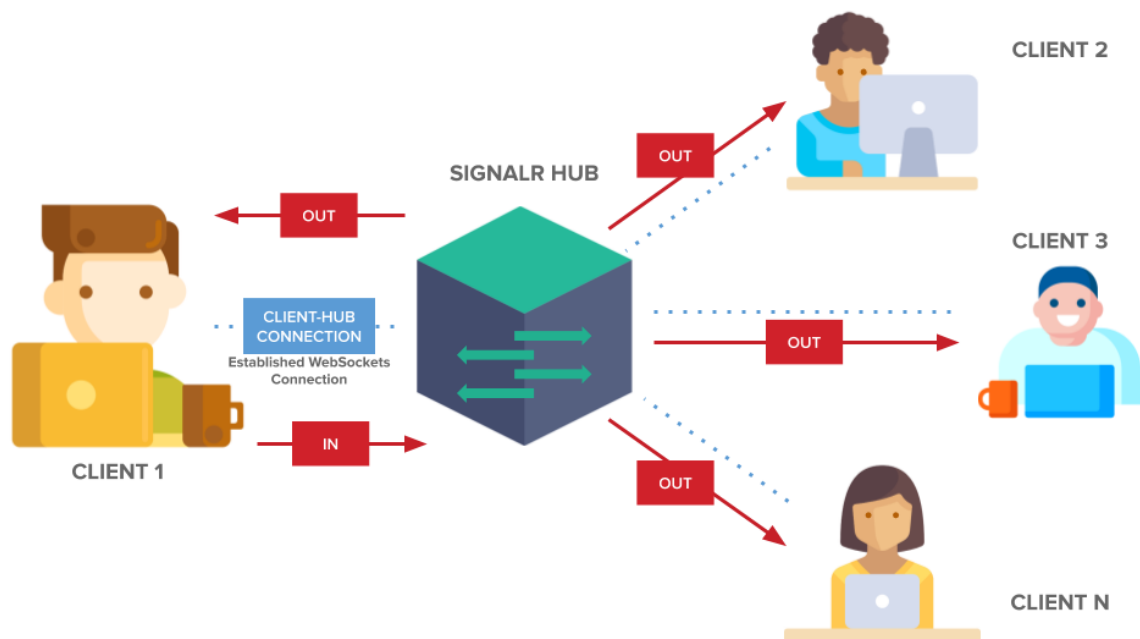


Figure 36 SignalR client-server communication (PubNub, 2019)

SignalR takes advantages of several transports and automatically selects the best available transport. A SignalR connection starts as HTTP and, if it is possible, is promoted to a WebSocket connection. WebSocket is the ideal transport for SignalR, but if WebSocket is not available, it falls back to other transports. The advantage of usage of the ASP.NET SignalR is that the application code is not dependent on transport type and a developer does not need to adapt the code to a specific transport. SignalR uses HTTP and connects server and clients through an open-connection that keeps connectivity open. The connectivity is terminated only if client loses online connectivity or physically terminates it.

ASP.NET SignalR contains two models for communication between clients and servers: Persistent Connections and Hubs. A Persistent Connection is a simple endpoint for sending and broadcasting messages. A Hub provides more high-level functionality, which is built

upon the Persistent Connection and allows clients and server to call methods on each other directly. In “CPS Agile” web application are used Hubs communication models.

7.2.4 Entity Framework

As it was described previous in the paper, EF is an open source ORM framework, which offers an automated mechanism for storing and accessing the data in the database. Entity Framework takes care of creating database connections, executing commands and automatically converting query results to the C# domain-specific objects. The role of Entity Framework in application’s architecture is illustrated in Figure 37.

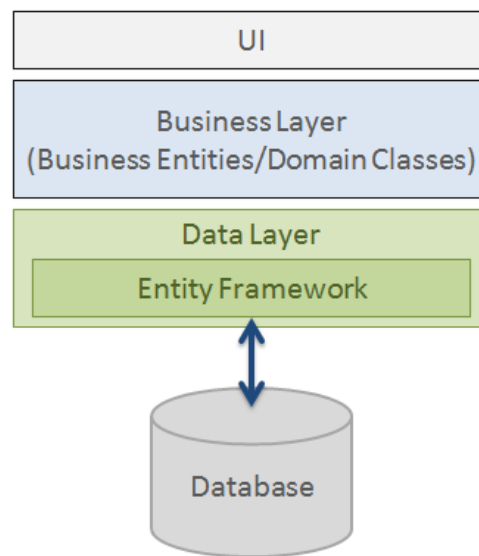


Figure 37 Entity Framework in application’s architecture (Entityframeworktutorial.net, 2013)

Entity Framework provides an abstract level to work with tables and columns by using the domain-specific objects. This approach increases code readability and eliminates the need to write the data-access code for manipulation with the database. SQL commands to the database can be executed using LINQ to Entities.

7.2.5 LINQ

Language Integrated Query (LINQ) is an SQL-like query syntax in .NET languages, which helps to retrieve data from different data sources, such as SQL database, ADO.NET dataset, object collections, XML documents and so on. LINQ is integrated in C# and provides a single querying interface for different types of data sources. Results of LINQ queries are objects. It enables to use object-oriented approach and eliminates the need of transforming different formats of results into objects (see Figure 38).

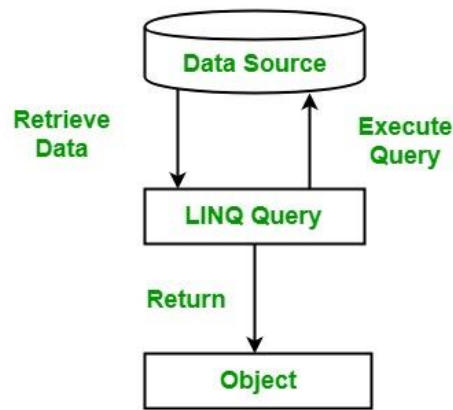


Figure 38 Retrieving data with LINQ (GeeksforGeeks, 2019)

In the “CPS Agile” web application LINQ is mainly used for working with object collections (LINQ to Objects) and retrieving data from the database with help of Entity Framework (LINQ to Entities). The example of retrieving a collection of User Stories from the database is illustrated in Code 6. Code 7 illustrates the usage of LINQ to Objects.

Code 6 Retrieve User Stories from the database using LINQ to Entities

```

1 public List<UserStory> GetUserStories(string pokerPlanningId)
2 {
3     using (var db = CpsContext.Create())
4     {
5         var userStoryList = db.UserStories
6             .Where(w => w.PokerPlanningId == pokerPlanningId).ToList();
7
8         return userStoryList;
9     }
10 }
  
```

Code 7 Delete existing exported Excel files using LINQ to Objects

```

1 public string ExportPokerPlanning(string pokerPlanningId)
2 {
3     // delete existing files
4     List<FileInfo> previousFiles = new DirectoryInfo(
5         HttpContext.Current.Server.MapPath("~/Temp/"))
6         .GetFiles("*PokerPlanning*").ToList();
7
8     previousFiles.ForEach(x => x.Delete());
9
10    . . .
11 }
  
```

7.2.6 EPPlus

EPPlus is an open-source .NET library that allows reading and writing Excel files using the Office Open XML format (xlsx). In the “CPS Agile” web application EPPlus library is used for exporting the results of the events to Excel files.

7.2.7 Frontend technologies

For implementation of the frontend part of the application were used such libraries and technologies as HTML5, CSS, jQuery, AJAX, Clipboard.js, Bootstrap and Font Awesome. Font Awesome is a font and icon toolkit based on CSS. The “CPS Agile” web application uses icons from this toolkit. Bootstrap is an open-source CSS framework, which contains different CSS and JS design templates for buttons, forms, navigation lists and other UI components. Clipboard.js allows copy text to clipboard without Flash. JQuery is an open-source JS library, which simplifies navigation of document, selection of DOM elements, creation of CSS animation and AJAX usage. AJAX is a set of web development techniques, which helps to create asynchronous web applications. AJAX decouples the presentation layer and data exchange layer, so that application can send and retrieve data from server asynchronously in the background. It allows to dynamically update a content on the web page without reloading it. The “CPS Agile” application uses AJAX, for instance, for deleting event from “My Events” web page (see Code 8).

Code 8 Delete event on Events.cshtml

```
1 $('#btn_cnfDelete').click(function() {
2     $('#div_text').hide();
3     $('#div_loader').show();
4     $('#btn_cnfDelete').prop('disabled', true);
5     $('#btn_cancel').prop('disabled', true);
6
7     $.ajax({
8         type: "POST",
9         url: '@(Url.Action("DeleteEvent", "Event"))' + "/" + itemToDelete,
10        success: function(data) {
11            var url = '@Url.Action("Events", "Event")';
12            window.location.href = url;
13        },
14        error: function(data) {
15            $('#div_failed').show();
16            $('#div_loader').hide();
17            $('#btn_cnfDelete').prop('disabled', false);
18            $('#btn_cancel').prop('disabled', false);
19        }
20    });
21 });
```


AJAX calls HTTP POST method *DeleteEvent* on the server side and then updates the web page, if event was successfully deleted.

7.3 Hosting

The implemented prototype of “CPS Agile” web application was published on the Microsoft Azure cloud computing platform. The “CpsAgileDb” database, which is used by the application, is hosted on the created server on the Microsoft Azure. The major benefits of hosting “CPS Agile” web application in the cloud are security, high availability and scalability. Application can be found on web address <http://cpsagile.azurewebsites.net/>.

7.4 Summary

For the implementation of “CPS Agile” prototype I used popular technologies and libraries, but the most important factor is that these technologies are used by virtual team, which is working on the case project. For that reason, the team can change and adapt the application to their needs at any time. The application is an open-source software, and code is stored in the public repository on GitHub. The “CPS Agile” web application and its database are situated on the cloud computing platform MS Azure and is accessible to the public on <http://cpsagile.azurewebsites.net/> web address.

8 Evaluation and future improvements

This chapter is dedicated to the final evaluation of the created artifact, the results and implications. Firstly, the Scrum Master of the case project executed the acceptance testing and the prototype was slightly modified according to the comments from the client. Then the “CPS Agile” web application was used by the virtual team to conduct Retrospective meeting, discuss the previous Sprint and determine what can be changed to make the next Sprint more productive. The application was also used to estimate features from the Product Backlog as the part of Sprint Planning ceremony. Based on the performed validation, list of limitations and list of future improvements were established.

8.1 Acceptance testing results

The developed application was firstly shown and consulted with the Scrum Master. After consultation with the Scrum Master, minor changes were introduced to the application. For the simplicity reason and more user-friendly interface it was decided by the client to change the way of working with the Retrospective items (aka sticky notes). In accordance with the use cases presented in the subsection 6.1.3 Retrospective Use Cases, adding, deleting and sharing actions were implemented with the modal window. After the first acceptance testing with the client, it was decided to simplify the work with points and replace modal window with a simple textbox, so users can work with the points directly on the main page. This change simplified the solution and made it more user-friendly. The change affected use cases RUC4, RUC5 and RUC7. The change did not have effect on the functional requirements, as the modification was about one of the possible ways of the requirements' implementation. The modification is presented in Figure 39.

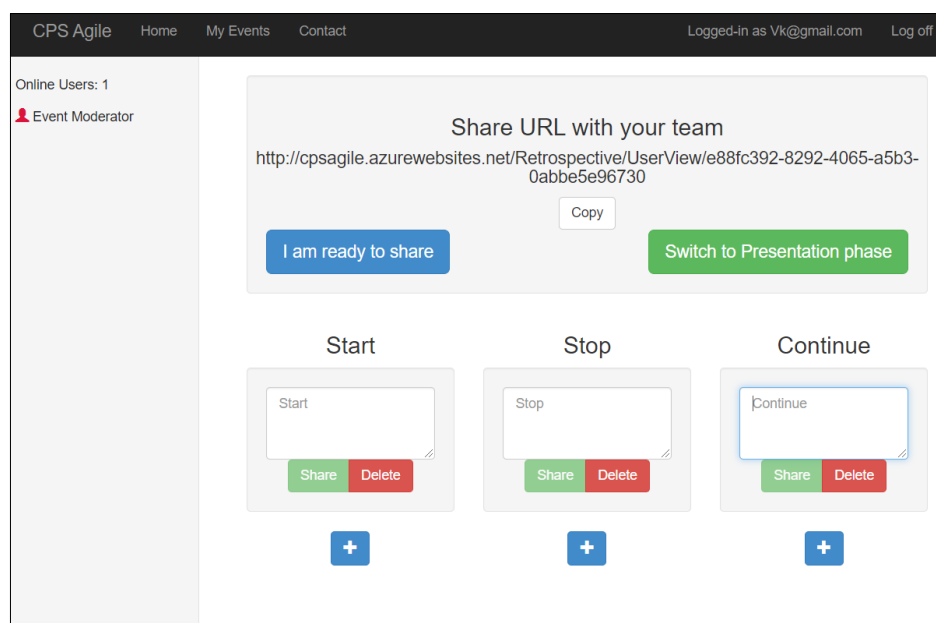


Figure 39 Modification in the work with the items (aka sticky notes) after the first testing

The same comment about modal window redundancy was made for PPUC3, creation of new User Story. It was decided to replace modal window with a simple textbox (see Figure 40).

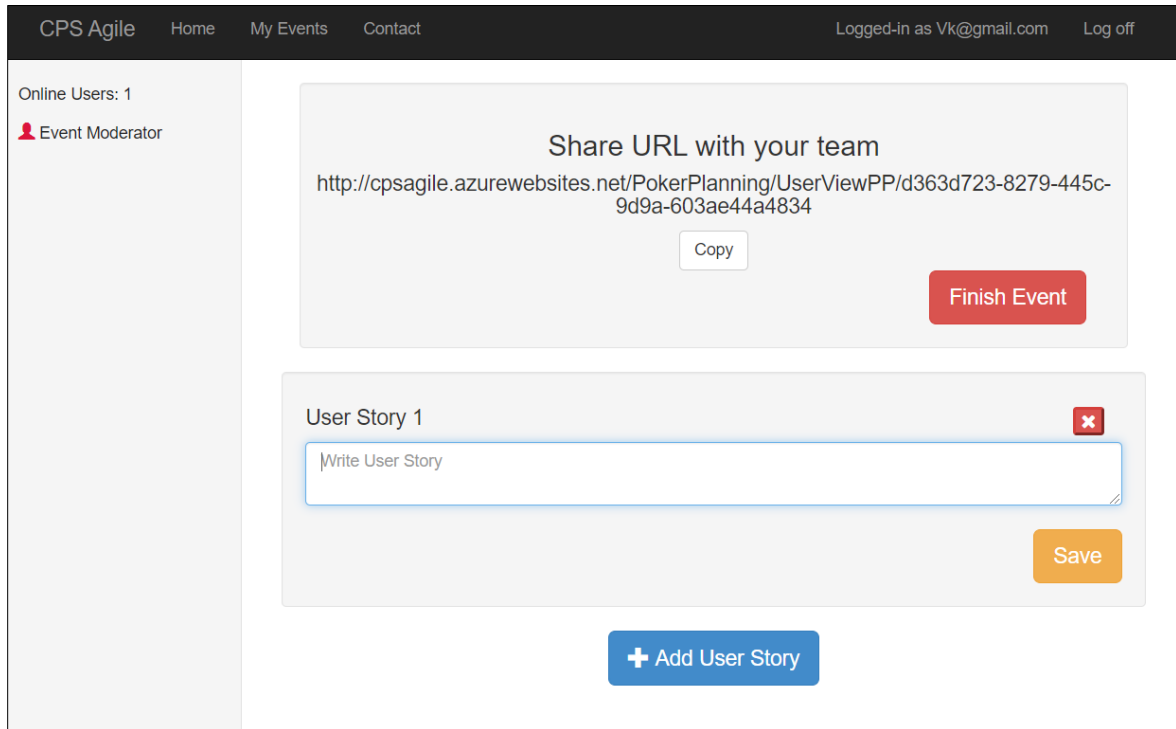


Figure 40 Modification in User Story creation

The acceptance testing with the client showed that the implemented PPUC6 needs to be changed, as the pop-up modal window distracts from the main purpose of the event – estimation activity. When one of the participants chooses the coffee card, the pop-up modal window is shown right after the transition to the Discussion phase. For that reason, the moderator cannot see the chosen estimations and has to decide immediately about the interruption of the event. If the moderator declines the suggestion, there is no way how he can later pause the event. And if the moderator accepts the suggestion and interrupts the event, the moderator sees the estimation results only after stopping the coffee break. This issue occurred due to mistakes in the initial design of the application and due to not precisely defined PPFR7 requirement from the client.

The issue was addressed by presenting a „Coffee break” button instead of pop-up window. If someone chooses the coffee card, then in the Discussion phase of the event the “Coffee break” button on the main panel occurs for the moderator and by clicking on this button, he or she pauses the activity and starts the coffee break. These modifications were accepted by the client and final graphical user interface can be seen in Figure 41. The change did not present modifications in the functional requirements, as it was only about one of the possible ways of the PPFR7 requirement implementation. By presenting the „Coffee break” button, the current workflow was adjusted, and the previous

inconveniences were addressed. Now the moderator can firstly discuss the estimation results with the team, and after the agreement with the participants to pause the event for coffee break.

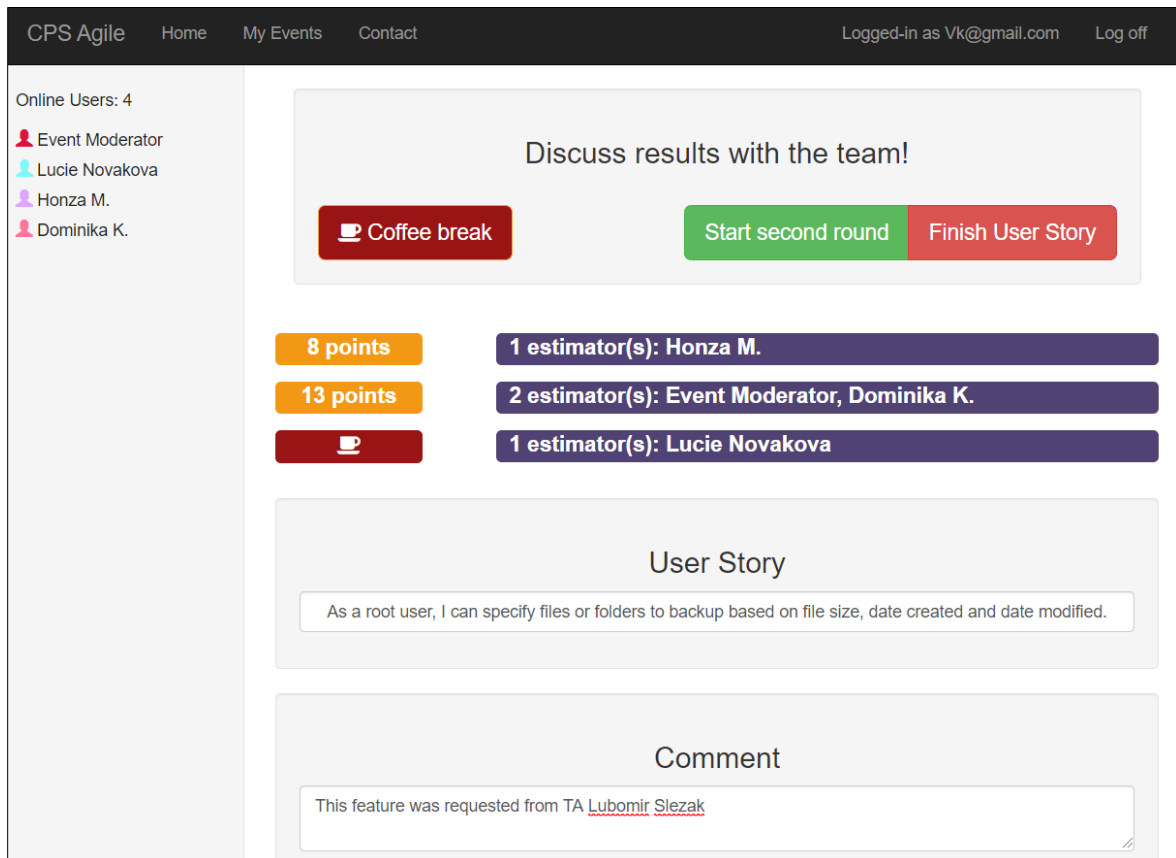


Figure 41 Modified "Coffee break" functionality

Another modification, which was made on the client's demand, was related to the Voting phase. After the first acceptance testing with the client, the idea to show to the user a number of available votes had emerged. When an event is transited to the Voting phase, the number of available votes for user would appear on the main panel. When a participant of the event votes for an item, the number of remaining votes on the main panel is reduced by one. This functionality was not listed in the final requirements on the application; therefore, it presented the change in the functional requirements on the Retrospective feature and created a new use case. The change in the graphical user interface is presented in Figure 42.

The acceptance tests also showed that there is no need in unique colorful name of each user as it overwhelms the user interface and is perceived as unprofessional and too polychrome. Therefore, the RUC3 was slightly adjusted, and the colorful figures next to the names were presented instead of colorful names on the panel with the list of participants. The modifications can be seen in Figure 42.

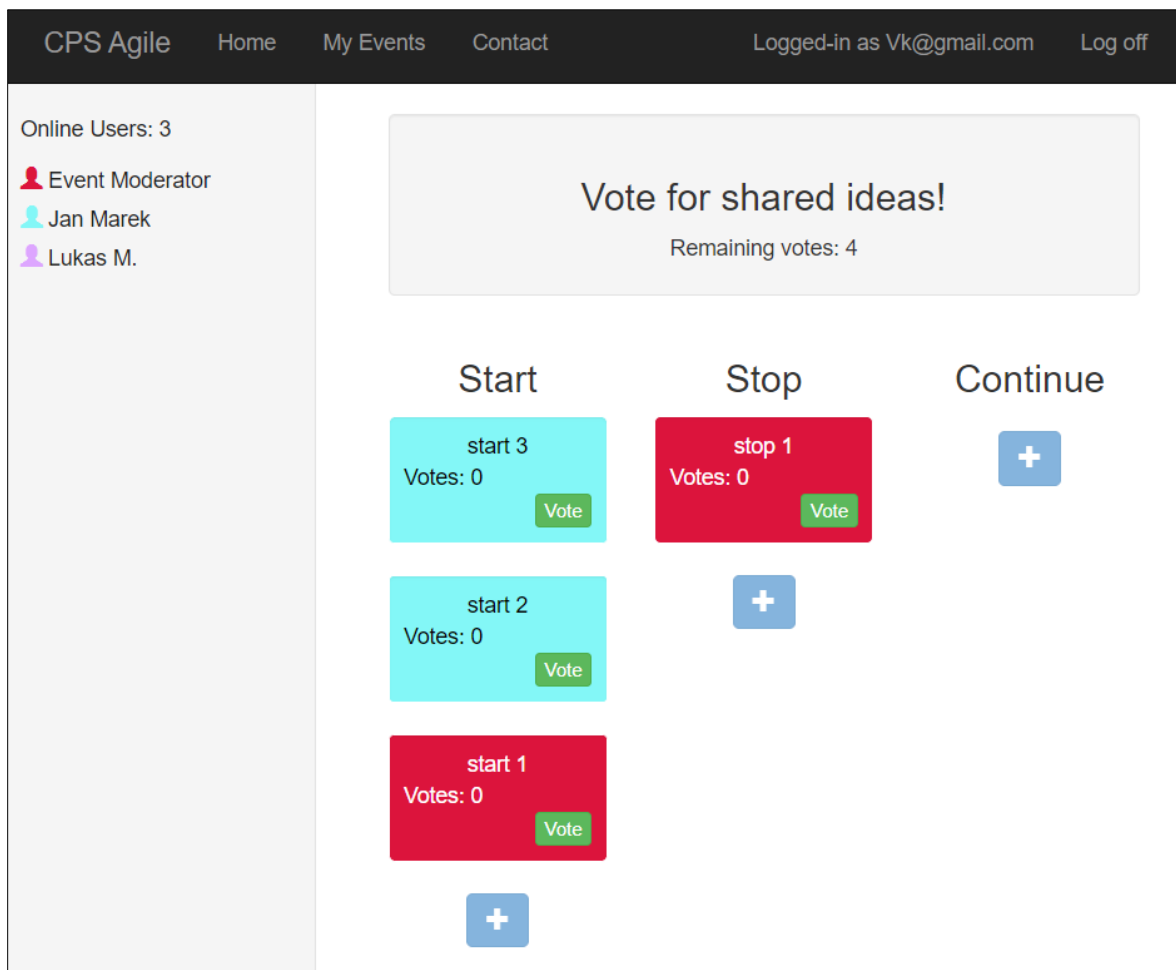


Figure 42 New requirement about Available votes, changes in GUI

After the minor modifications, which were described above, the client accepted the solution and the “CPS Agile” web application was used to conduct Retrospective and Sprint Planning Scrum ceremonies on the case project. The main results and implications are described in the following section.

8.2 Main implications

This section comments on how the created artifact solves common challenges, which are related to the virtual work environment, and which were present on the case project.

In comparison to the previous state of agile processes and the way, how the ceremonies were carried before introducing the “CPS Agile” artifact, the time, spent on the planning and Retrospective meetings, was dramatically reduced. This change is explained by the fact, that the “CPS Agile” solution presents out-of-the-box features and prepared environment for these processes. The solution eliminated the need to transfer the retrospective points from remote participants to sticky notes or the final results of ceremonies to an electronic form. As the result, costs of maintenance of team knowledge were reduced.

Due to some sort of the process gamification, the team members, especially who was remotely connected to the event, got more engaged in the meetings. Expanded level of interest and involvement resulted in the increased perception of remote team members as a part of the team. It contributed to building a cohesive team culture and helped to operate with a common purpose.

As the “CPS Agile” solution contains several agile practices, it helps to maintain the knowledge in one place and facilitates knowledge sharing. That approach makes traceability more efficient and solves knowledge fragmentation problem, which is common for virtual work environment (Lous et al., 2018).

The “CPS Agile” artifact contributed to a creation of work environment, where physical presence of team members is not required to participate in particular Scrum activities. This approach helps to reduce the costs of running the virtual team, as meeting rooms with costly video-conferencing systems are not required for these agile practices.

One of the main contributions of “CPS Agile” application laid in the fact, that the Product Owner started to participate in Retrospective events. The Product Owner, who did not physically attend Retrospective event before, could now connect to the meetings remotely, comment and reflect on the ideas. Participation of the Product Owner, as a stakeholder’ representative, contributed to the transparency among sites and reduced the gap between the Development Team and stakeholders.

The “CPS Agile” solution was tested on the case project in a bank environment. However, the web application is not limited to the specific case project or financial environment and can be used in different industries, not only in a software development field. Created software artifact presents out-of-the-box features and can be used by any agile team. The application can be used by small and middle teams, but will be the most valuable for virtual teams, that does not share the same physical location, as it digitalizes certain agile techniques and practices.

8.3 Limitations

The main limitations are presented by the technologies, which were used to implement the “CPS Agile” application prototype.

As it was stated in the Implementation chapter, the SignalR framework uses different transport types, but the most preferable is the WebSocket transport, as it has the lowest latency and the most efficient use of server memory. Nevertheless, there are specific requirements on the server, in order to use WebSocket transport for SignalR framework. The server must fulfill requirements of using .NET Framework 4.5 and Windows Server 2012 or Windows 8, in order to be able to provide WebSocket transport. The “CPS Agile” web application is hosted on the Microsoft Azure Web Sites, where I had to enable WebSocket support specifically in the configuration on Azure platform.

SignalR imposes a strict limitation on IIS web server. IIS on Windows 8 or Windows 7 has a limit of ten simultaneous connections. The limit is reached very quickly as the connections

are frequently re-established and are not disposed immediately after the usage. Therefore, I used the IIS Express version during the implementation of “CPS Agile” web application, as the IIS Express version does not have this limit. SignalR framework also imposes the limitation on the applications regarding version of jQuery library. The applications must use jQuery version 1.6.4 or higher.

Another limitation is requirements on the clients. Microsoft provides a list of browsers, which support SignalR framework (Microsoft, 2018):

- Microsoft Internet Explorer versions 8, 9, 10, and 11.
- Mozilla Firefox: current version - 1, both Windows and Mac versions.
- Google Chrome: current version - 1, both Windows and Mac versions.
- Safari: current version - 1, both Mac and iOS versions.
- Opera: current version - 1, Windows only.
- Android browser

But in addition to the required certain version of browser, the various HTML5 transports, which are used by SignalR, have their own requirements, which are presented in Table 5.

Table 5 Web Browser Transport Requirements (Microsoft, 2018)

Transport	Internet Explorer	Chrome	Firefox	Safari	Android
WebSockets	10+	current - 1	current - 1	current - 1	N/A
Server-Sent Events	N/A	current - 1	current - 1	current - 1	N/A
ForeverFrame	8+	N/A	N/A	N/A	4.1
Long Polling	8+	current - 1	current - 1	current - 1	4.1

Besides technical requirements, there are also limitations, which are brought by the way of prototype's implementation. In the current version of “CPS Agile” web application only the moderator can reconnect to the existing event more times. The unique identifier for a regular user is the connection id, which is generated, when client connects to a hub. For that reason, when regular user closes a browser tab or refreshes a web page with the event, he loses the connection, and the hub assigns new connection id to that user. This fact can be extremely inconvenient, especially during the Preparation phase of Retrospective session, when in case of reconnection a regular user loses all his or her prepared items. This problem can be addressed by changing the way of mapping user to the connection. The current version of “CPS Agile” web application uses *in-memory storage* approach, specifically a dictionary structure, which represents a collection of keys and values. This approach enables to get list of currently connected users and ensures optimal performance, because dictionary structure is implemented in C# as a hash table and retrieves the value by its key with $O(1)$ time complexity. In order to persist information after restarts, *in-memory storage* approach can be changed to *permanent, external storage*, such as a database table or Azure table storage for storing information about connections.

Another current limitation of “CPS Agile” application is, that if moderator reconnects to the Retrospective event during the Point Preparation phase, the information about prepared points is lost. The reason for that behavior is the current implementation of workflow. This restraint can be eliminated by changing business logic of the application and saving prepared items to a database already at the Point Preparation phase.

There is a limitation of my paperwork, which is presented by the nature of case study. The single case study presents the generalizability limitation of the results. The results are grounded on the analyzed case project and are affected by the specific context, specifically recent adaptation of agile practices on the large enterprise project. There is a possibility, that the created artifact will not present the same effective results on the projects, where the tools and agile processes are already strongly established and settled. Nevertheless, the created “CPS Agile” artifact is easy to change and adopt to the current needs, due to its solution architecture, which separates business logic, data and presentation layers. For that reason, the developed solution can be transformed and used as well in another contexts. In addition, the detailed information of the context was provided in the paper, which allows future studies to replicate or compare the results.

To ensure reliability of the research and to reduce the risk of single-researcher bias, the open communication with the representatives of case project team was kept during the whole process of research. This approach contributed to a constant verification of the research direction and reduced the risk of misinterpretations.

8.4 Suggestions for future improvements

This section describes future improvements, which can be later applied to the created “CPS Agile” artefact. A list of future improvements is a result of suggestions and recommendations, which were made by the virtual team, after conducted planning and Retrospective meetings using the application.

One of the first suggestions was to switch to HTTPS and adopt SSL (TLS) protocol to ensure communications security. There are several advantages of using HTTPS over HTTP. Firstly, the usage of TLS certificate prevents from such attacks as BGP hijacking, DNS hijacking, domain spoofing and man-in-the-middle attacks. Secondly, the usage of Internet security protocols increases search ranking, as search engines favor HTTPS websites over HTTP. Moreover, users trust the website more, when they see a green padlock in the browser’s address bar.

The next suggestion was about buying a custom domain, in order to shorten the web address. However, few team members express the opinion that *cpsagile.azurewebsites.net* domain is suitable and easy to remember, and a minor complication is undesired advertising of Azure services.

Few suggestions were made regarding enabling registration and login using social networks. Allowing external authentication via social networks is highly specific and dependent on the corporate culture (for example, some financial institutions block access to social

networks on workstations). Therefore, this feature was not implemented in the first version of “CPS Agile” application.

Other features of “CPS Agile” application, which were not implemented in the current version, but can be adopted to the client’s needs, is an email confirmation to complete registration and password reset via email. In the current version a password can be changed only through account settings. The “CPS Agile” application also supports two-factor authentication but does not implement this feature for the case project.

One significant suggestion was made regarding an account of registered user. It was recommended to add a username field to the account, so this username is then displayed in the list with currently connected users, and user’s role as a moderator is shown next to the username in round brackets. The current solution displays only a role, Event Moderator, in the list with connected users, and adds an unwanted anonymity to the event. The change will be implemented in the next version of “CPS Agile” application, where username field will be added to a registration form and will be required in order to complete registration.

Another suggestion came from the Product Owner after carrying Poker Planning ceremony. The Product Owner would welcome the enhancement of “coffee break” feature, specifically the ability to set a certain timeframe for coffee break duration. It was proposed, when the moderator starts coffee break, he can also set how many minutes a coffee break will last. Then a modal window, which is shown to every participant, contains a countdown clock, which shows how many minutes till the event continues. The timer will be an optional feature and time setting will not be explicitly required to start a coffee break. The requested functionality will be implemented in the next version of “CPS Agile” web application.

After carrying few Retrospective meetings there came a preposition to change business logic in the voting process. It was purposed by few members of the Development Team to allow to vote for one retrospective point more than once. The concept of total assigned votes will remain the same, so a regular user can decide to give all his or her available votes to only one specific point. This change in business logic will be considered later by the business. Additionally, an idea about enhancing “My Events” page occurred, so a registered user can click on a particular event on “My Events” page and event details will be displayed in new page. This functionality will introduce visualization of the outcomes and eliminate the need to download an Excel file, in order to see the results of past events.

In next versions of “CPS Agile” application, I would like to introduce improvement from the technical side, specifically I am planning to include Dependency Injection Container (IoC Container) 7.1to the solution, in order to automatically manage dependencies and object creation.

8.5 Summary

The article “Virtual by Design” (Lous et al., 2018) affirms that a careful selection of the practice-tool combination increases the effectiveness on the project. As the result of the “CPS Agile” artifact application on the large enterprise project in a bank environment,

the time, which is spent on the planning and Retrospective meetings, was reduced. Due to some sort of the process gamification, the team members, especially who was remotely connected to the event, got more engaged in the meetings, so as the result a cohesive team culture was facilitated. The created artifact facilitates such dimensions of cooperation, as communication, trust, collaboration, awareness and coordination. Despite the single case nature limitation, the “CPS Agile” web application is not limited to the case project or financial environment. Its Retrospective and Poker Planning estimation techniques can be used in different industries, not only in software development field.

The created software artifact presents out-of-the-box features and can be used by any agile team. The application will be the most valuable for virtual teams, who does not share the same physical location, as it digitalizes certain agile techniques and practices. Due to unique assignment of colors to users and real-time nature of the application, the “CPS Agile” is suitable for the large self-organizing teams. Export functionality of “CPS Agile” application provides the data about each Retrospective point, including information about who in the team came up with the idea and proposed the item. The “CPS Agile” web application supports several the most popular agile practices, so it reduces knowledge fragmentation problem and maintains team knowledge in one place. The created “CPS Agile” software artifact facilitates transparency in Scrum team, reduces the gap between stakeholders and the Development Team, and encourages each team member to fully participate in the agile processes regardless of the different physical location. The virtual team on the case project evaluated the application as the enhancement of current agile processes, and the client is ready to support the application by providing resources for graphical user interface and future improvements, which were described in section 8.3.

Conclusions

The case project of this Master thesis has recently adopted agile project management methodology, namely Scrum framework. Before transition to the agile methodology the large enterprise project with a virtual team was managed using waterfall model. Scrum framework is oriented on small co-located teams; therefore, Scrum ceremonies and other agile practices were hold on the project with certain challenges and impediments. For that reason, the need to change and improve current agile processes occurred. In order to improve current state of agile practices on the project, the artifact "CPS Agile" web application was created. The "CPS Agile" web application digitalized current agile processes, namely Retrospective and Poker Planning Scrum ceremonies. The application was used by the team to carry planning and Retrospective meetings. The results showed that the usage of "CPS Agile" web application reduced time, which was spent on the meetings, and changed the meetings in a sort of gamification and friendly way. The team members got more engaged in the meetings, which positively reflected on the building relationships inside geographically dispersed team and facilitated a cohesive team culture. The Product Owner, who did not physically attend Retrospective event before, could now participate in the meetings remotely, comment and reflect on the ideas. Participation of the Product Owner, as a stakeholder' representative, contributed to reducing the gap between the Development Team and stakeholders. Digitalization of Scrum practices positively reflected on the overall team's attitude to the transition, as they evaluated the artifact as the advancement of current agile processes.

Objective fulfilment

The main objective of this Master thesis was to develop a web application, which contributes to agile project management practices and improves agile processes on the large enterprise project with a virtual team. The main goal was fulfilled by creating the artifact – "CPS Agile" web application. The goal was achieved by completing the individual smaller objectives such as analysis of problem area and current state of the agile practices on the case project, requirements establishment, detailed design and the development process. The solution was tested and validated on the large enterprise project in a bank environment. To reach the main goal was used Design Science Research methodology, which was described in the second chapter.

Before the physical implementation of the application, I conducted a research on the problem area, using Systematic Literature Review methodology. The outcome of research was reported in the first chapter of my paperwork. The results were divided into three groups based on the resource type – Research articles, Bachelor and Master theses, Specialized literature.

The third chapter was dedicated to the theoretical description of agile project management practices, their differences from traditional models, and illustration of core specifics of Scrum framework. The current state of agile methodologies worldwide was presented using statistics from the State of Agile annual report.

The fourth chapter detailed current agile practices on the case project, their weaknesses and limitations. Then the research of existing competing solutions was made, which showed that there is no implemented solution, which would fit all requirements of the Scrum team on the case project.

The fifth chapter was devoted to the establishment of requirements from users. As nowadays probably the most important thing in the implementation of business applications is to meet very complex user needs. For this reason, I held the kickoff meeting with the members of case team, where the general requirements were discussed, and the name of the application was chosen. Then I held consultations with project managers and Scrum Masters, who are engaged in agile project delivery, as they are expected to be the main users who decide whether they introduce the application on the project. At the end of the chapter, the final list of requirements on the application arose and was approved by the client.

In the sixth chapter the design of “CPS Agile” web application was presented by introducing use case diagrams with actors and main activities, solution’s architecture and the database model in a form of physical entity relationship diagram. The coverage of requirements was verified by provided traceability matrixes. Then the draft of graphical user interface was created, consulted with the main users and after the revision approved by the contracting authority.

After the completion of application design, the implementation of “CPS Agile” solution was described in the seventh chapter. The chapter thoroughly illustrated architecture of the application, used frameworks and technologies. The solution was implemented using MVC ASP.NET technology together with SignalR framework, which takes care of real-time nature of the application. The “CPS Agile” web application was published on the cloud computing platform Microsoft Azure and is accessible to the public on <http://cpsagile.azurewebsites.net/> web address. The application is a free and an open-source software, which provides the possibility to adapt and change functionality according to the team’s current needs.

The last chapter is dedicated to validation of the developed solution, its application on the large enterprise project and the description of main results. It was presented how the “CPS Agile” artifact addresses challenges, which are specific to a virtual work environment. The developed solution facilitates such dimensions of cooperation, as communication, trust, collaboration, awareness and coordination. The created “CPS Agile” artifact proved, how the smart adaptation and alignment of agile practices in combination with the practical tool, which is well-accepted by the team, can support agile processes on the global software development project.

List of references

Agile Alliance. (2001). *12 Principles Behind the Agile Manifesto* | Agile Alliance. [online] Available at: <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>.

Agilemanifesto.org. (2001). *Manifesto for Agile Software Development*. [online] Available at: <https://agilemanifesto.org/> [Accessed 31 Oct. 2019].

Bass, J.M. (2014). *Scrum Master Activities: Process Tailoring in Large Enterprise Projects*. 2014 IEEE 9th International Conference on Global Software Engineering.

Brown, D.M. (2011). *Communicating design: developing web site documentation for design and planning*. Berkeley, Ca: New Riders.

Buchalceková, A. and Stanovská, I. (2013). *Příklady modelu analýzy a návrhu aplikace v UML*. Praha: Oeconomica.

Ca.gov. (2017). *Waterfall and Agile* | Project Resources. [online] Available at: <https://projectresources.cdt.ca.gov/agile/waterfall-and-agile/> [Accessed 31 Oct. 2019].

CollabNet VersionOne. (2013). *7th Annual State of Agile Survey*. [online] Available at: <https://www.stateofagile.com/#ufh-i-338592786-7th-annual-state-of-agile-report/473508> [Accessed 30 Oct. 2019].

CollabNet VersionOne. (2015). *9th Annual State of Agile Survey*. [online] Available at: <https://www.stateofagile.com/#ufh-i-338592786-7th-annual-state-of-agile-report/473508> [Accessed 30 Oct. 2019].

CollabNet VersionOne. (2019). *13th Annual State of Agile Survey*. [online] Available at: <https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508> [Accessed 30 Oct. 2019].

C-sharpcorner.com. (2019). *Understanding SignalR From Scratch*. [online] Available at: <https://www.c-sharpcorner.com/article/understanding-signalr-from-scratch/> [Accessed 3 Nov. 2019].

Dyba, T. and Dingsoyr, T. (2015). *Agile Project Management: From Self-Managing Teams to Large-Scale Development*. 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering.

Entityframeworktutorial.net. (2013). *What is Entity Framework?* [online] Available at: <https://www.entityframeworktutorial.net/what-is-entityframework.aspx> [Accessed 4 Nov. 2019].

Entityframeworktutorial.net. (2019). *DbContext in EF 6*. [online] Available at: <https://www.entityframeworktutorial.net/entityframework6/dbcontext.aspx> [Accessed 3 Nov. 2019].

Excentia.es. (2019). *What is a traceability matrix? - TraceabilityX for Jira - Confluence*. [online] Available at: <https://confluence.excentia.es/pages/viewpage.action?pageId=81133892> [Accessed 22 Sep. 2019].

Garrett, J.J. (2011). *The elements of user experience: user-centered design for the Web and beyond*. Berkeley, Ca: New Riders.

GeeksforGeeks. (2019). *What is Query in LINQ? - GeeksforGeeks*. [online] Available at: <https://www.geeksforgeeks.org/what-is-query-in-linq/> [Accessed 4 Nov. 2019].

Grombířová, A. (2019). *Analýza dopadů implementace metodiky SCRUM v bankovním prostředí*. Diplomová práce. Vysoká škola ekonomická v Praze. Vedoucí práce Jan Pour. [online]. Available at: <https://theses.cz/id/bnuzza/> [Accessed 19 Oct. 2019].

Guid.one. (2019). *What is a GUID*. [online] Available at: <http://guid.one/guid> [Accessed 24 Sep. 2019].

Holmström, H., Fitzgerald, B., Ågerfalk, P.J. and Conchúir, E.Ó. (2006). *Agile Practices Reduce Distance in Global Software Development*. Information Systems Management, [online] 23(3), pp.7–18. Available at: <https://www.brian-fitzgerald.com/wp-content/uploads/2013/10/Holmstrom-et-al-2006-Agile-GSD.pdf> [Accessed 26 Sep. 2019].

Horváthová, M. (2019). *Analysis of agile software development methodology usage in work teams*. Bakalářská práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Barbora Bühnová. [online]. Available at: <https://is.muni.cz/th/cg5sv/> [Accessed 19 Oct. 2019].

Hubený, M. (2018). *Škálování agilního vývoje*. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Tomáš Pitner. [online]. Available at: <https://is.muni.cz/th/px4ms/> [Accessed 19 Oct. 2019].

Iivari, J. and Venable, J. (2009). *Action research and design science research - Seemingly similar but decisively dissimilar*. 17th European Conference on Information Systems. Available at: https://www.researchgate.net/publication/221407297_Action_research_and_design_science_research_-_Seemingly_similar_but_decisively_dissimilar [Accessed 19 Oct. 2019].

Ingebrigtsen, E. (2015). *SinalR, real-time application development*. Packt Publishing.

Johannesson, P. and Perjons, E. (2014). *A Method Framework for Design Science Research*. An Introduction to Design Science, pp.75–89.

Kalenda, M. (2017). *Scaling Agile Software Development in Large Organizations*. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Bruno Rossi. [online]. Available at: <https://is.muni.cz/th/yg3vg/> [Accessed 19 Oct. 2019].

- Kaushik, P. (2016). *A Detailed Look at 3-Tier Software Architecture*. [online] Techopedia.com. Available at: <https://www.techopedia.com/2/32100/software/a-detailed-look-at-3-tier-software-architecture> [Accessed 1 Nov. 2019].
- Kiesel, M. (2012). *Softwarové prostředky pro podporu virtuální projektové spolupráce*. Diplomová práce. Masarykova univerzita, Filozofická fakulta. Vedoucí práce Zdeněk Kadlec. [online]. Available at: <https://theses.cz/id/59pdeq/> [Accessed 19 Oct. 2019].
- Krotkov, A. (2018). *Adoption of agile approaches to large IT projects*. Diplomová práce. Vysoká škola ekonomická v Praze. Vedoucí práce Jan Kučera. [online]. Available at: <https://theses.cz/id/pd6en5/> [Accessed 19 Oct. 2019].
- Lous, P., Tell, P., Michelsen, C.B., Dittrich, Y., Kuhrmann, M. and Ebdrup, A. (2018). *Virtual by design*. Proceedings of the 13th Conference on Global Software Engineering - ICGSE '18.
- Mathiassen, L. (2000). *Object oriented analysis et design*. Aalborg, Denmark: Forlaget Marko.
- Microsoft (2018). *Supported Platforms*. [online] Microsoft.com. Available at: <https://docs.microsoft.com/en-us/aspnet/signalr/overview/getting-started/supported-platforms> [Accessed 10 Nov. 2019].
- Mihalache, A. (2017). *Project Management Tools for Agile Teams*. Informatica Economica, 21(4/2017), pp.85–93.
- Nekola, O. (2012). *Scrum proces v distribuovaném týmu*. Bakalářská práce. Vysoká škola finanční a správní. Vedoucí práce Jan Lánský. [online]. Available at: <https://is.vsfs.cz/th/mfx3q/> [Accessed 19 Oct. 2019].
- Neon Rain Interactive. (2015). *Using Agile Scrum for Web Development | Neon Rain Interactive*. [online] Available at: <https://www.neonrain.com/agile-scrum-web-development/> [Accessed 31 Oct. 2019].
- Nerur, S., Mahapatra, R. and Mangalaraj, G. (2005). *Challenges of migrating to agile methodologies*. Communications of the ACM, [online] 48(5), pp.72–78. Available at: <https://dl.acm.org/citation.cfm?id=1060712> [Accessed 24 Oct. 2019].
- Nyrud, H. and Stray, V. (2017). *Inter-team coordination mechanisms in large-scale agile*. Proceedings of the XP2017 Scientific Workshops on - XP '17.
- Peppers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). *A Design Science Research Methodology for Information Systems Research*. Journal of Management Information Systems, 24 (3), pp. 45–77.
- PubNub. (2019). *What is SignalR?* [online] Available at: <https://www.pubnub.com/learn/glossary/what-is-signalr/> [Accessed 4 Nov. 2019].
- Razi et al. (2018). *Cloudemy: Step into the Cloud*. [online]. Available at: <https://dl.acm.org/citation.cfm?id=1060712> [Accessed 4 Nov. 2019].

Scrumalliance.org. (2019). *About Us*. [online] Available at: <https://www.scrumalliance.org/about-us>.

Scrum.org. (2019). *The Scrum Guide*. [online] Available at: <https://www.scrum.org/resources/scrum-guide> [Accessed 30 Oct. 2019].

SearchSoftwareQuality. (2019). *What is use case? - Definition from WhatIs.com*. [online]. Available at: <https://searchsoftwarequality.techtarget.com/definition/use-case> [Accessed 20 Sep. 2019].

Tutorialsteacher.com. (2019). *Introduction IoC, DIP, DI and IoC Container*. [online] Available at: <https://www.tutorialsteacher.com/ioc/introduction> [Accessed 3 Nov. 2019].

Unicorn. (2019). *About us*. [online] Available at: <https://unicorn.com/company-profile> [Accessed 14 Aug. 2019].

Usman, M., Britto, R., Damm, L.-O. and Börstler, J. (2018). *Effort estimation in large-scale software development: An industrial case study*. Information and Software Technology, 99, pp.21–40.

Vaishnavi, V., & Keuchler, W. (2004). *Design Research in Information Systems*. Association for Information Systems. Available at: <http://desrist.org/desrist/article.aspx>. [Accessed 19 Oct. 2019].

Venable, J. and Baskerville, R. (2012). *Eating our own Cooking: Toward a More Rigorous Design Science of Research Methods*. Electronic Journal of Business Research Methods, 10 (2), pp. 141–153.

Visual-paradigm.com. (2018). *What are Scrum Artifacts?* [online] Available at: <https://www.visual-paradigm.com/scrum/what-are-scrum-artifacts/> [Accessed 31 Oct. 2019].

Visual-paradigm.com. (2019). *What is Entity Relationship Diagram (ERD)?* [online] Available at: <https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/> [Accessed 24 Sep. 2019].

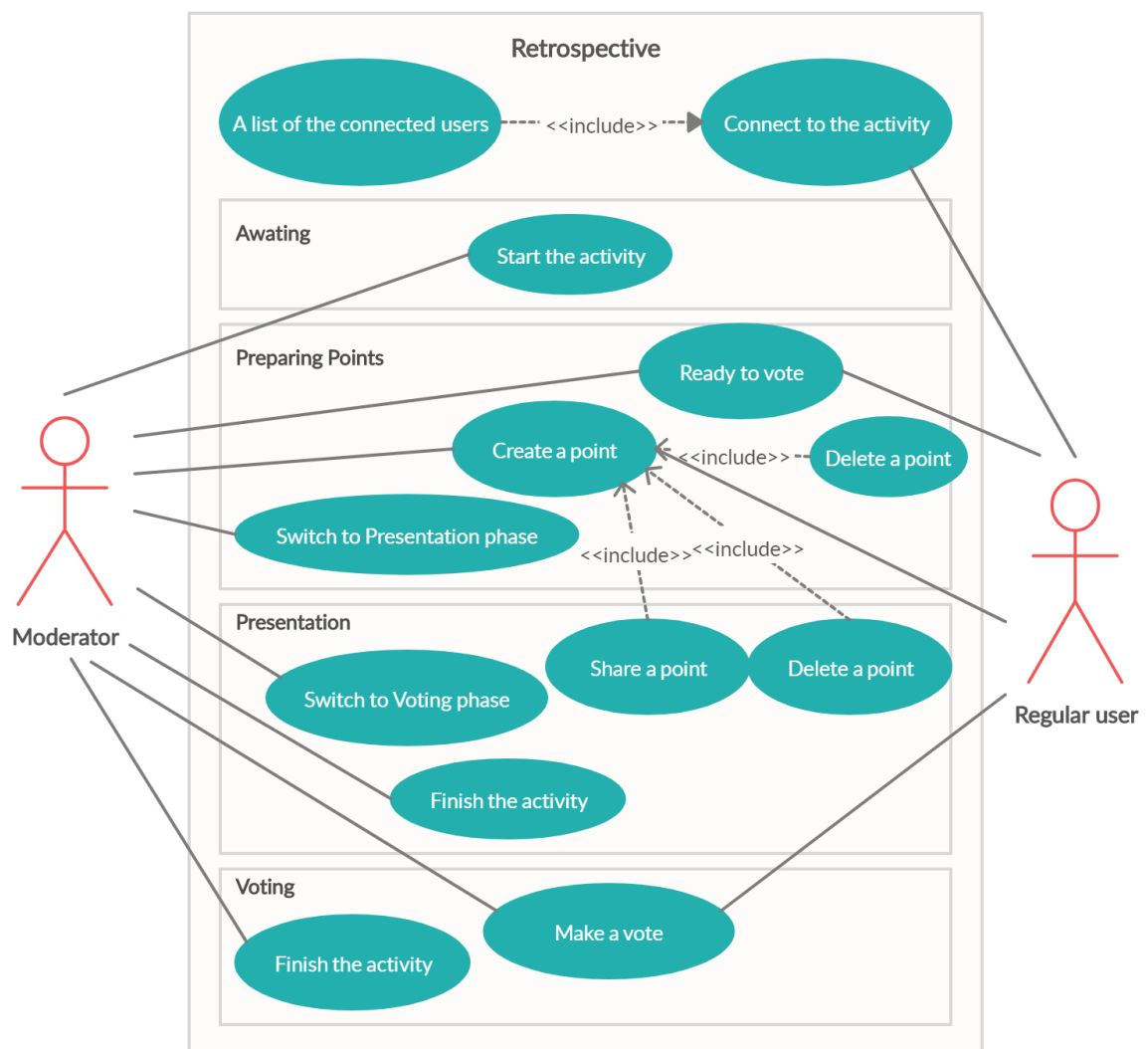
Annexes

Annex A contains UML Use Case Diagrams for all three features of the “CPS Agile” application – the Retrospective, the Poker Planning and the Brainstorming features.

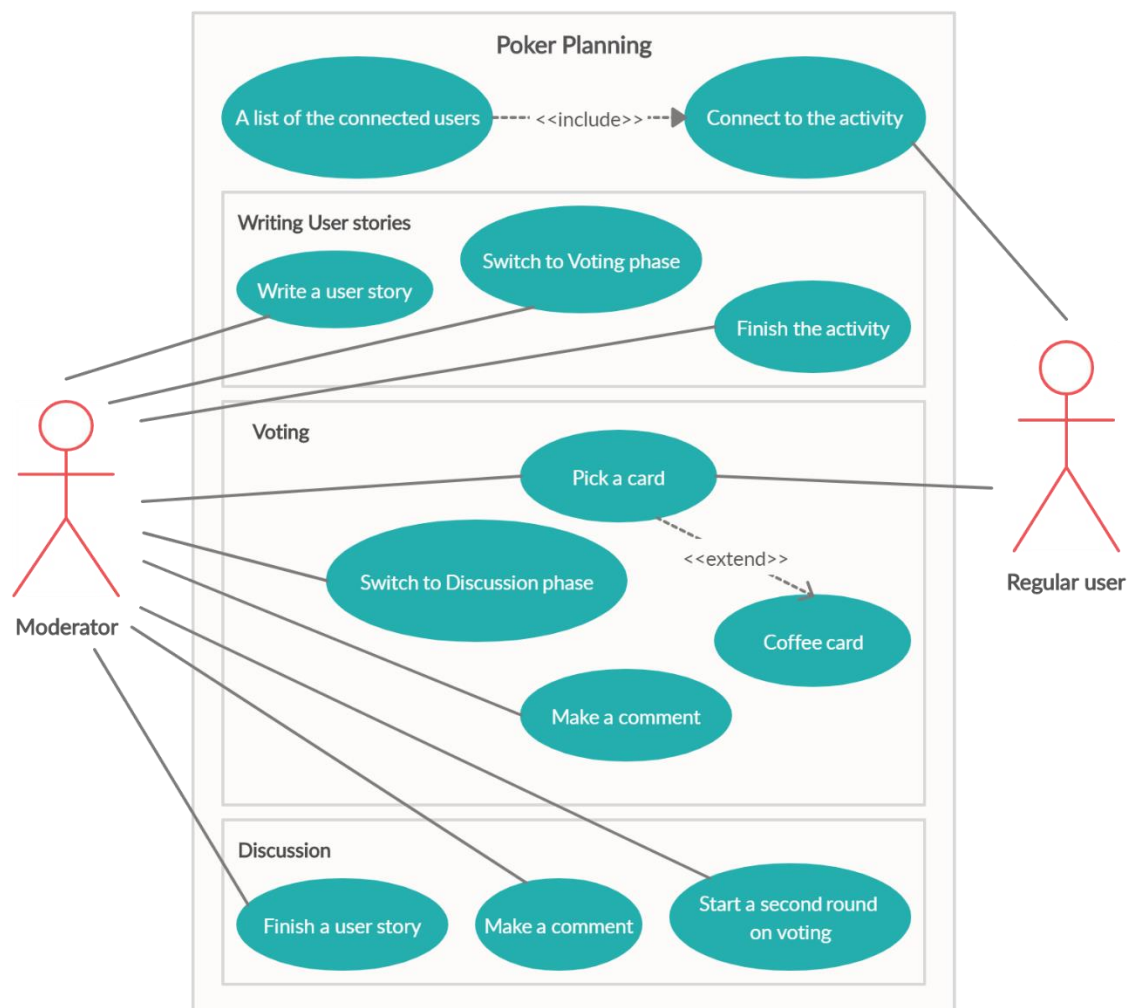
Annex B contains complete Entity Relationship Diagram, created in the Enterprise Architecture software.

Annex A: UML Use Case Diagrams

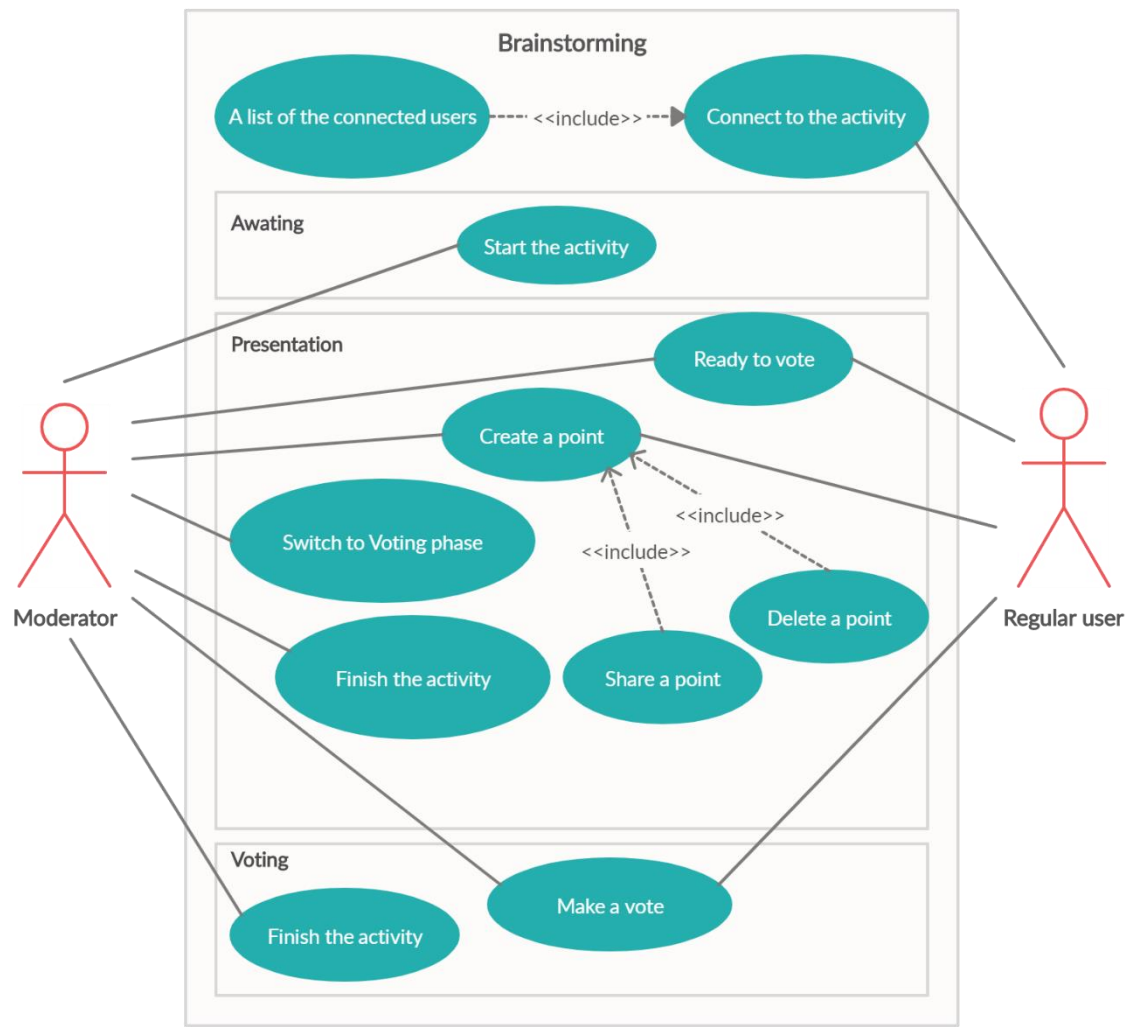
Retrospective Use Case Diagram



Poker Planning Use Case Diagram

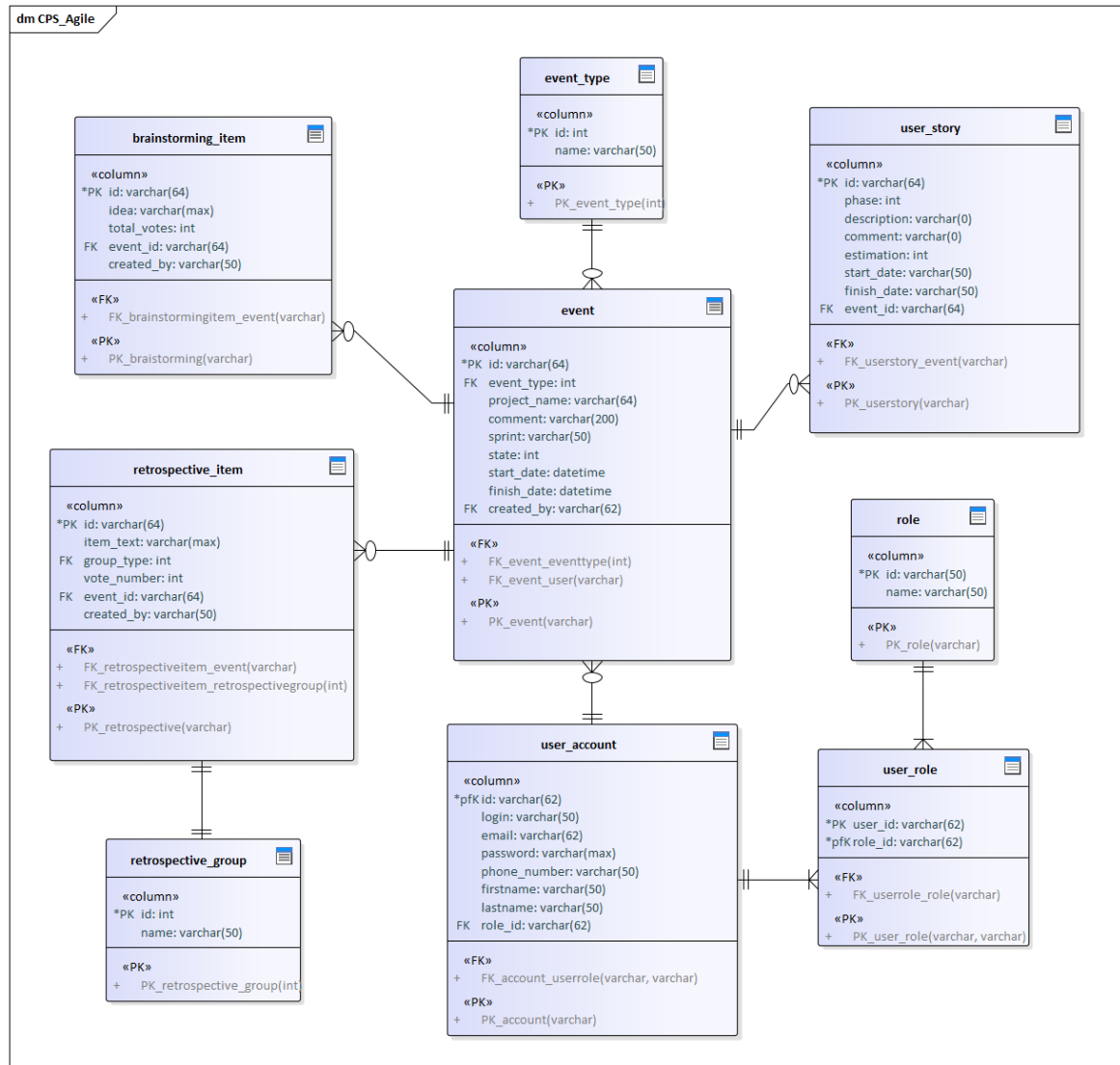


Brainstorming Use Case Diagram

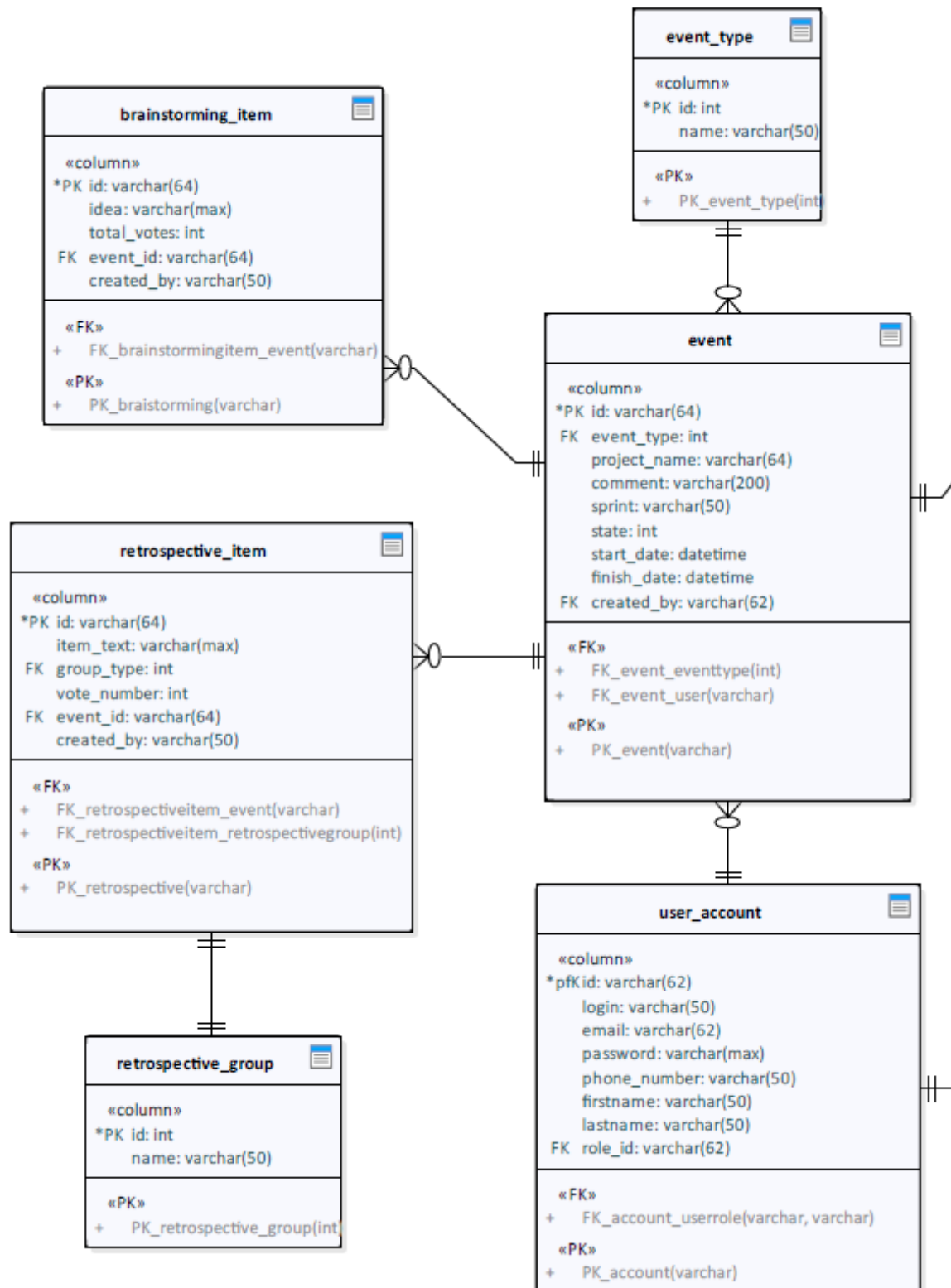


Annex B: Physical Entity Relationship Diagram, created in EA

Complete ERD



Entity Relationship Diagram, Part 1



Entity Relationship Diagram, Part 2

