National College of Ireland

Object Oriented Software Engineering
Project Part 2

Student Name(s): Jacob Benarros Neto, Aurimas Jurgutis

Student ID(s): x20208731, x20182104

Date of Submission: 6th of May 2021

Lecturer: Victor Del Rosal

Table of Contents

Task 8 – Risks, Quality and Communication

Risk management:

The 'Online Booking System' project will use an Agile approach to manage risks. It is important to identify risk in advance, so that the Scrum master can implement contingency plans to eliminate or decrease risk damage to the project. Firstly, it is important to define risk to better manage them.

Understanding risks

This project will design an efficient approach to manage risks based on the literature of the concept of risks. The project will provide "the optimum allocation of resources according to the impact and probability of (…) risks." (Gellert, 2015, p. 13). Risks normally rely on risk management methods evolving from economic cost-benefit analysis and scientific methodologies of risk assessment (Black and Baldwin, 2010). This project will analyse risks based on its probability of occurrence and its severity, similar to the template scale below:

			Potential Consequences				
			L6	L5	L4	L3	L2
			Minor injuries or discomfort. No medical treatment or measureable physical effects.	Injuries or illness requiring medical treatment. Temporary impairment.	Injuries or illness requiring hospital admission.	Injury or illness resulting in permanent impairment.	Fatality
			Not Significant	**Minor**	**Moderate**	**Major**	**Severe**
Likelihood	Expected to occur regularly under normal circumstances	**Almost Certain**	**Medium**	**High**	**Very High**	**Very High**	**Very High**
	Expected to occur at some time	**Likely**	**Medium**	**High**	**High**	**Very High**	**Very High**
	May occur at some time	**Possible**	**Low**	**Medium**	**High**	**High**	**Very High**
	Not likely to occur in normal circumstances	**Unlikely**	**Low**	**Low**	**Medium**	**Medium**	**High**
	Could happen, but probably never will	**Rare**	**Low**	**Low**	**Low**	**Low**	**Medium**

GBKSoft (2021). Risks Involved In Development: Is It Worth Taking A Risk? https://gbksoft.com/blog/risks-involved-in-development-is-it-worth-taking-a-risk/

The risk-based approach is popular because it provides an evidence-based element that confers a rational decision-making power to control events (Rothstein, Huber and Gaskell, 2006). However, it is impossible to control risks completely (Gellert, 2018).

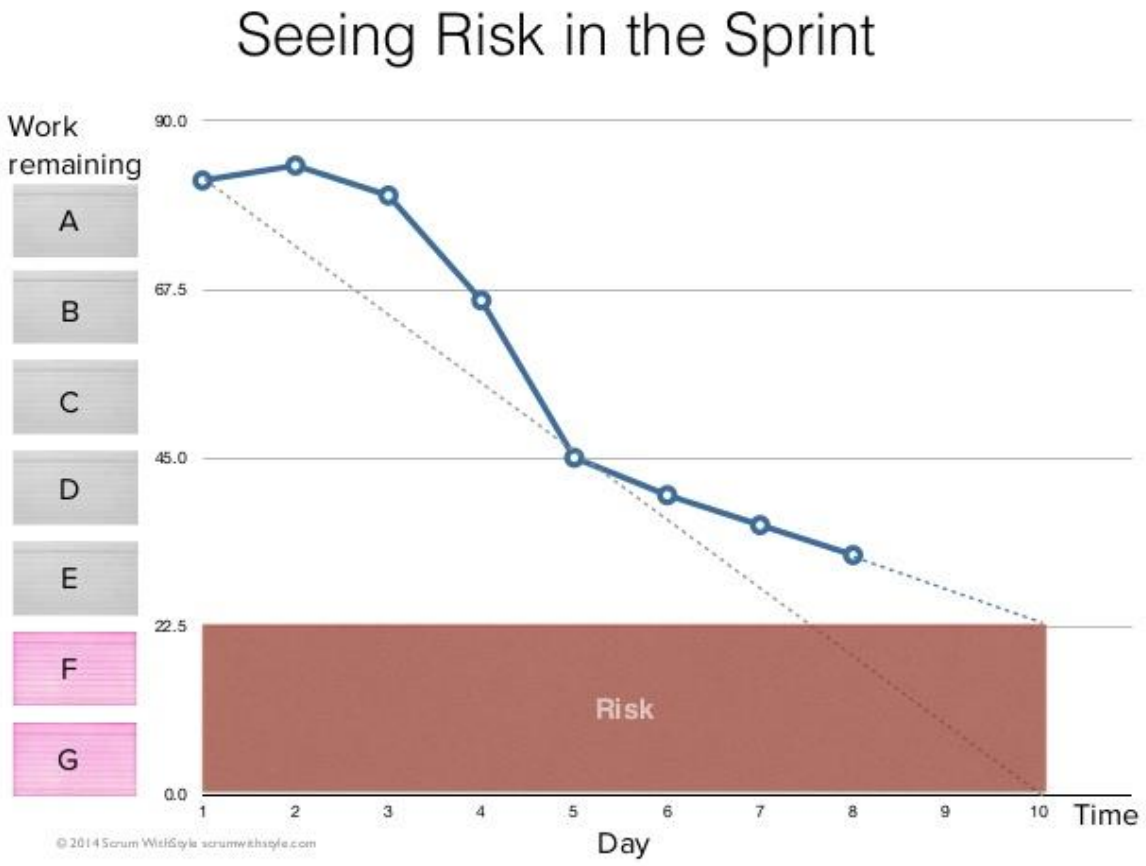
Risk is a human creation to control uncertain events. Gellert (2015) defines risk as a tool that uses statistics and probabilities to convert unpredictable, dangerous events into certain ones. Therefore, the author concluded that risks are unnatural elements; they are just methods to turn events into risks. Nothing is a risk, and at the same time anything is a risk; everything depends on how dangerous someone considers an event (Ewald, 2019).

Ad infinitum risk loop

"Risk is a knowledge-intensive concept" (Gellert, 2015, p. 15). That means risk factors are infinite: it depends on how much data humans can gather; the more people accumulate knowledge about risks, the more risk factors they can unravel (Gellert, 2015). That is an incessant search for knowledge that the author defines as information *ad infinitum,* contributing to *paralysis by analysis.* The limited concept of risks is incompatible with the idea that administrative and financial resources are limited (Harremoes *et al.*, 2013). However, this project will not intend to eliminate all potential risks but to control and decrease its damage when possible.

Risks are subjective

Risks are inevitably a subjective concept (Gellert, 2015); thus, individuals can interpret and value specific risks differently. That is the reason this project will use the Agile methodology to negotiate what risks the team will decide to monitor and manage. They team often re-evaluate risks priorities in every sprint backlog as the figure shows:

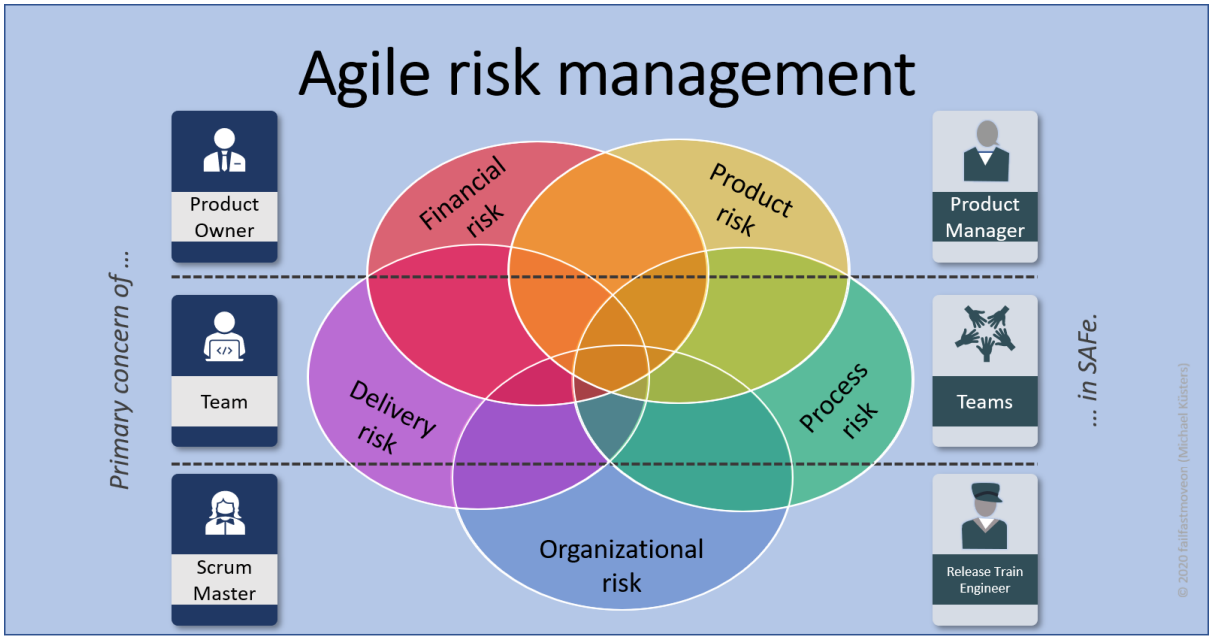


Source: Agile Risk Management (slideshare.net)

Managing Risks using Agile

Risks will be considered working items included in the Sprint or product backlog. They will be often updated and re-evaluated to account the changes, e.g. in the markets or new technologies.

For instance, traditional projects focus on foreseen risks concerning TBQ, i.e., time, budget, and quality. However, this project intends to encompass a broader approach considering internal and external factors, such as financial, product process, organisational, delivery risks. The figure below demonstrates how this project will approach risks:



Source: FailFastMoveon (2020). Agile Risk Management.
https://failfastmoveon.blogspot.com/2020/09/agile-risk-management.html

Furthermore, the Agile technique will not avoid the accumulation of risks. The project will be frequently tested for improvement at each iteration. That would not be possible if the project followed a plan driven approach, such as the waterfall methodology:

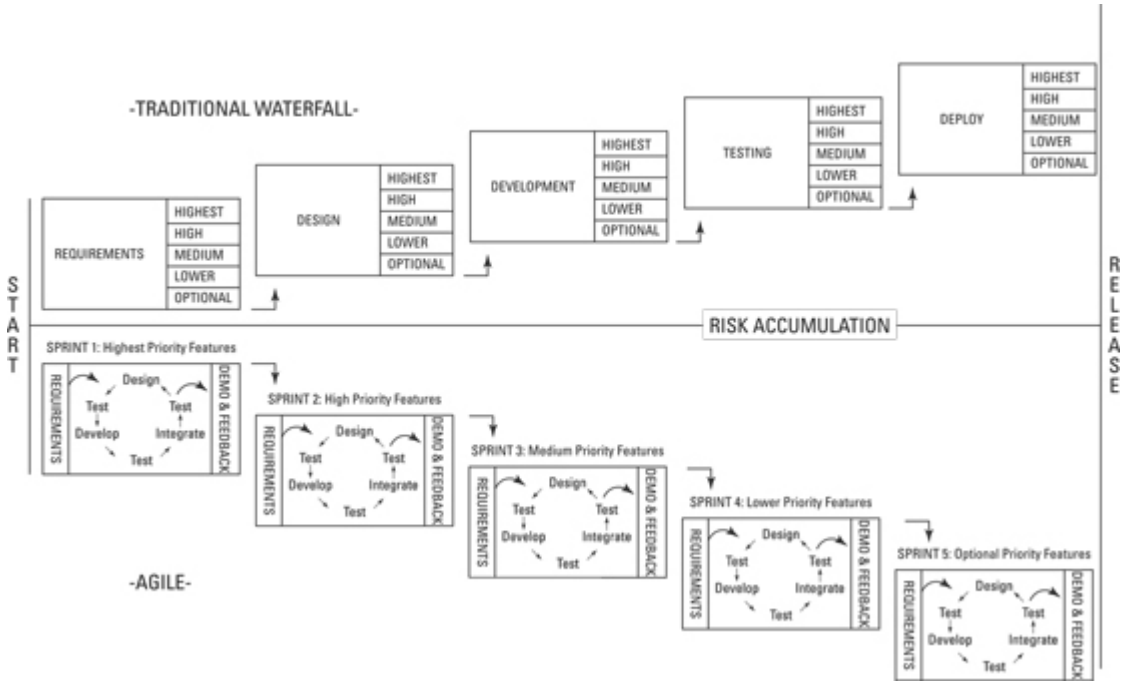
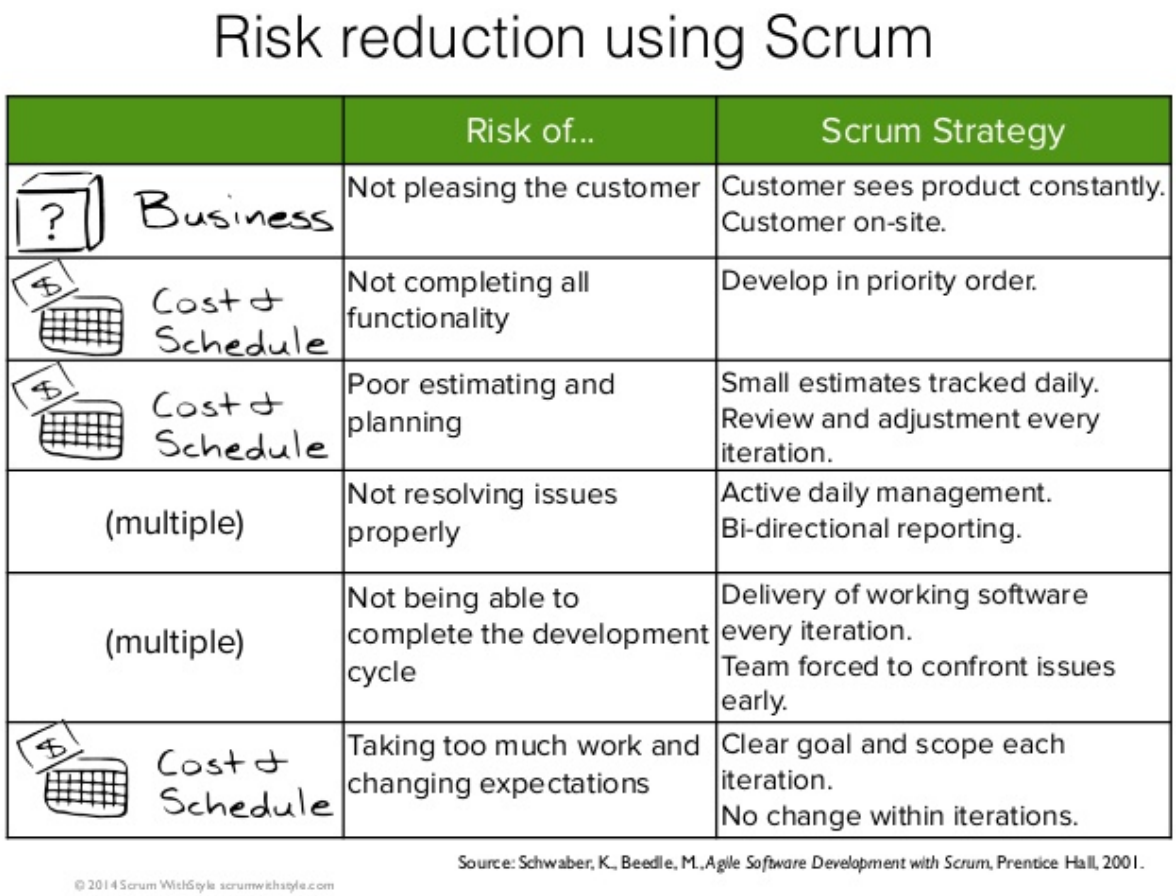


Figure 1 Risk accumulation: Agile vs Waterfall

Risks

It is therefore important to get an overview of the risks involved with developing our book publishing software. Here is a list of some of the biggest risks that this project may face when coding the software:

Source: [Agile Risk Management (slideshare.net)]



Source: [Agile Risk Management (slideshare.net)]

Business risk:

This project will use UML diagrams to communicate with customers to ensure the development team understands the customer product requirements. It allows the team to negotiate in advance what is feasible product to be delivered. Furthermore, it provides a product with expected results decreasing the probability of not pleasing the customer. Moreover, the project will follow the Agile methodology, that means that each deliverable will be changed several times in sprints, so that the customer will be able to provide feedback on functionalities he values the most.

Social risk:

The Scrum master must ensure that he can deliver the project with resources available the expected requirements. For instance, if he cannot hire highly skilled developers or if the technology is not advanced enough for producing the deliverables, he needs to communicate that vary clearly and limit the project scope.

Tech risk:

Poor Quality Code

Poor quality of code can cause several issues. For example, the code can be difficult to read for other developers to make changes or updates, or it can be rushed and then released without being tested, which leads to errors and bugs. To improve the quality of code, it is important to keep it to a good standard. This can be done in a few different ways:

1. Reviewing code with your team or with an experienced professional
2. Testing code before release
3. Working with coding standards and coding guides
4. Adding comments inside code files

Cost and Schedule Risks:

Time Constraints

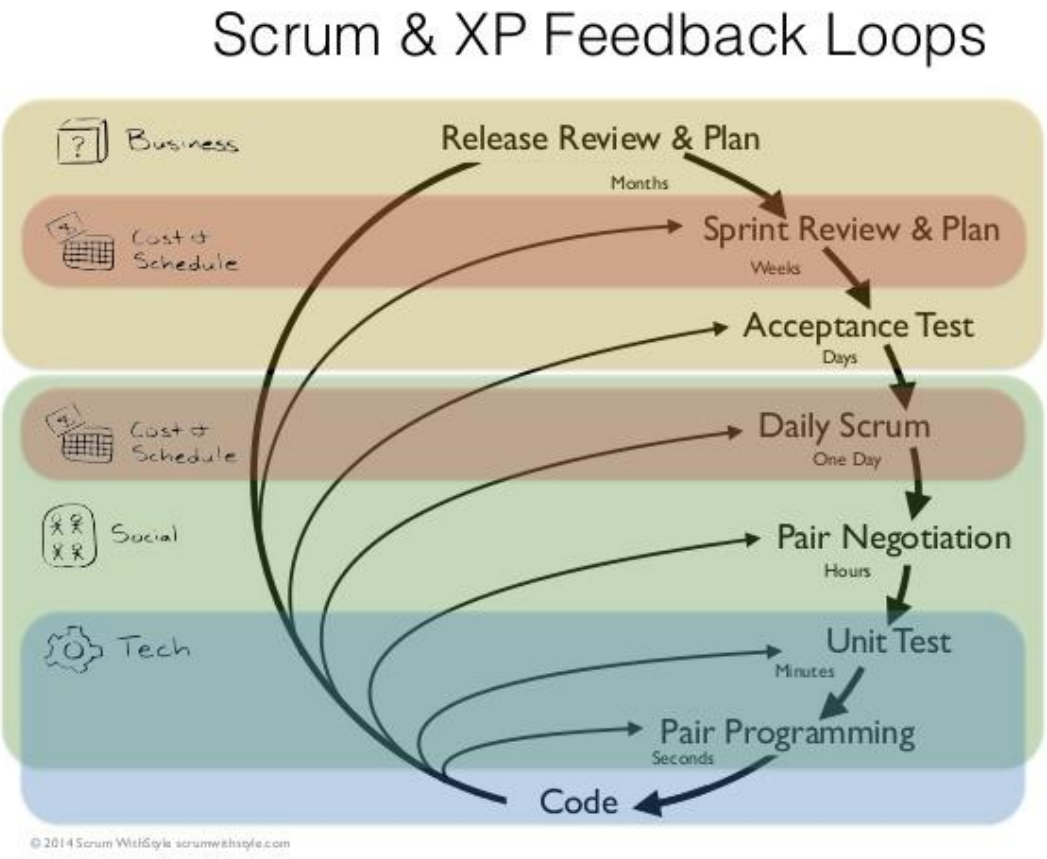
When developers are given a short amount of time to develop their code, the development of the program will become rushed, and it could lead to many issues. For instance, the program can be bug-ridden, unstable, and unusable after release, leading to dissatisfaction, or disproval, of customers/users. While this can be related to business projects, time constraints can also affect the project due to time mismanagement and leaving work up to the last few days of a deadline for that project decreasing quality assurance and increasing costs due mismanagement. Some of the ways to minimize risk of delaying the project are:

1. Manage time effectively by correctly defining the project and the methodology. That will avoid wasting time on pointless research and doing unnecessary tasks.
2. Breakdown the work and provide reasonable deadlines for smaller tasks i.e. giving enough time to write the project report.
3. Avoid meetings without a purpose can save valuable working time of the group.

This project will manage the risks presented using a Scrum approach, similar to the figure below:

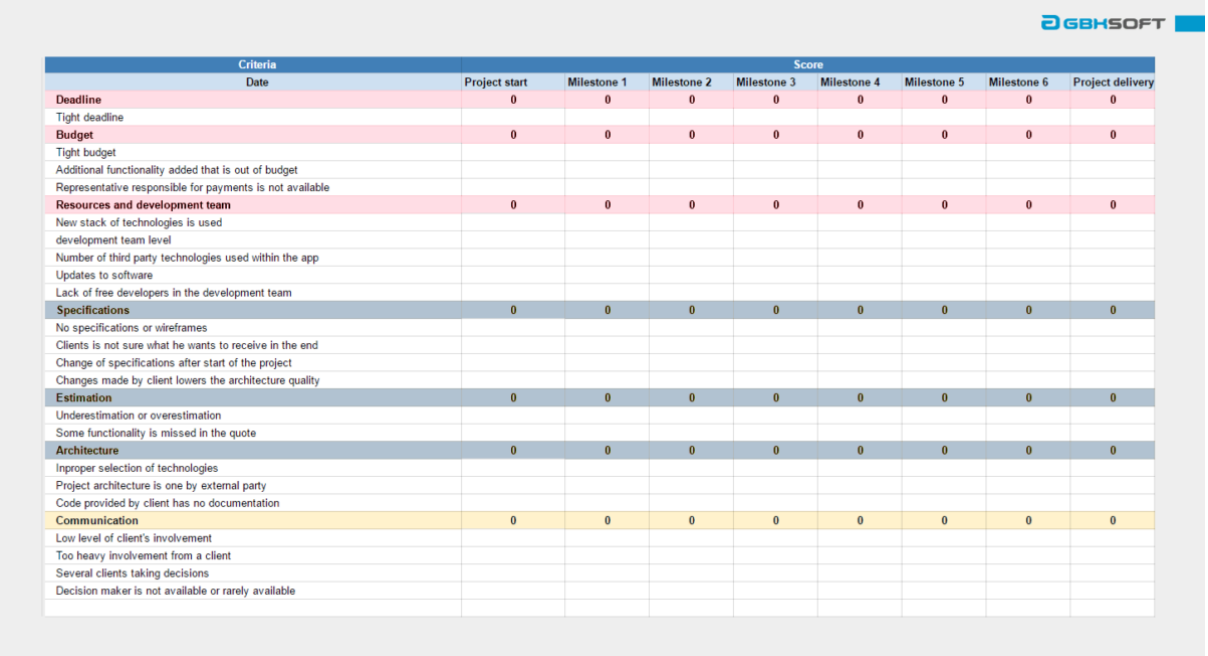
Source: [Agile Risk Management (slideshare.net)]

The team decided to use the Scrum process to approach risks because of its iterative approach to review existing and add new risks to the sprint backlog. The figure below presents how this project will implement that:



Source: [Agile Risk Management (slideshare.net)]

The ‘Online Booking System’ will design table similar the one below to mention the risk criteria, such as deadline, budget, resources, communication, and other risks. Then, it will note the score from the start, during its milestones until the delivery of the project.



Criteria	Score							
Date	Project start	Milestone 1	Milestone 2	Milestone 3	Milestone 4	Milestone 5	Milestone 6	Project delivery
Deadline	0	0	0	0	0	0	0	0
Tight deadline								
Budget	0	0	0	0	0	0	0	0
Tight budget								
Additional functionality added that is out of budget								
Representative responsible for payments is not available								
Resources and development team	0	0	0	0	0	0	0	0
New stack of technologies is used								
development team level								
Number of third party technologies used within the app								
Updates to software								
Lack of free developers in the development team								
Specifications	0	0	0	0	0	0	0	0
No specifications or wireframes								
Clients is not sure what he wants to receive in the end								
Change of specifications after start of the project								
Changes made by client lowers the architecture quality								
Estimation	0	0	0	0	0	0	0	0
Underestimation or overestimation								
Some functionality is missed in the quote								
Architecture	0	0	0	0	0	0	0	0
Inproper selection of technologies								
Project architecture is one by external party								
Code provided by client has no documentation								
Communication	0	0	0	0	0	0	0	0
Low level of client's involvement								
Too heavy involvement from a client								
Several clients taking decisions								
Decision maker is not available or rarely available								

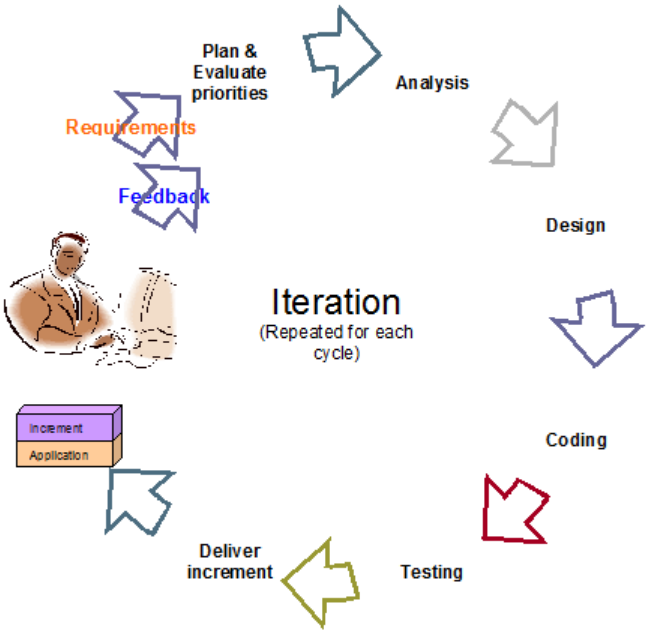
Source: [Risks in Web & Mobile Application Development [a Full Guide] | GBKSOFT]

Communication

Communication between developers and software engineers working on a project is important since ideas need to be delivered in some fashion for them to understand the project in detail before proceeding with coding. In our case, we used UML to portray the online book publishing system, what would be used in coding the program, and its expected outputs. To ensure that communication stays regular and consistent throughout the course of the project, meetings on Microsoft Teams would be set up once every week to ensure that our ideas and opinions are conveyed and applied, and UML diagrams are kept simple and to the point.

Quality

This project will ensure that the final product will achieve high quality standard because it will be reengineering project deliverables frequently. That is aligned the iterative Agile sprint backlog process that will perform incremental improvements. The deliverables will be tested and accommodate feedback from other stakeholders.



Source: Lethbridge/Laganière 2005.Chapter 11: Managing the Software Process

The Quality of Code

As discussed previously, keeping a good quality of code is important as it makes an impact on the overall quality of software. Good quality code is one that is simple, clear, bug-free and has a high level of performance. Good quality of code in software is important for many reasons:

1. The software becomes more robust; implementing understandable error messages for users to enable self-troubleshooting and indicate possible fixes for known errors

2. Ease to read and edit; keeping code simple and clear, inserting comments and clear notations can facilitate the work of other developers when editing code.

3. Easy to transfer the code across other operating systems/platforms. That improves the code standard so that developers don't have to rewrite it. Furthermore, it may cause compatibility issues for newer operating systems.

Quality of Software

The quality of software can be described in many ways. From a software engineers' perspective, it is a practice which improves the standard of the software, i.e. making/improving GUI, making the software simple and easy to use, and making the software pleasing to the eye.

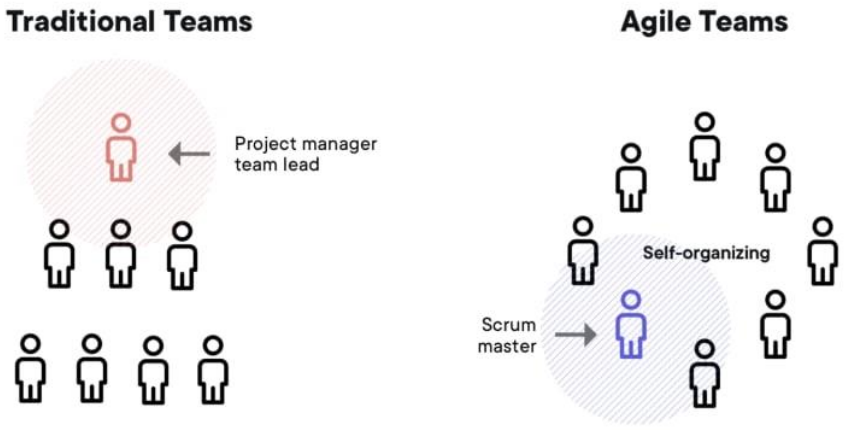
Communication within the team

Inefficient communication among stakeholder is itself a risk for the project. So, the team will ensure that communication, both oral and written is clear. Stand up meeting will be short. of 15 to 30 min maximum, and informative, e.g., promoting knowledge sharing and updates for this project. That is aligned with the Agile approach. The scrum master will ensure that project information and updates are widely and readily available to all stakeholders. Moreover, the project will use technology to encourage exchange of information among team members. Communication among members of the development team will use Microsoft Teams for meetings, PowerPoint, Gdoc, Google drive. Also, it will use Visual Paradigm to create UML diagrams ensuring 'non tech savvy' customers can communicate project requirements efficiently with software developers. Those diagrams will provide a clear understanding of that the customer requires before coding irrelevant software functionalities that increases the risk of delaying the project schedule and compromising resources and the budget.

Having good communication skills is an important factor in software engineering. The project will develop a system communicate efficiently and effectively following the Agile methodology. For instance, it will hold short team meetings to allow stakeholders to communicate their ideas, break up deliverables into smaller tasks and set deadlines, to share and ask feedback on UML diagrams/flowcharts projects. Furthermore, those meeting will be relevant to brainstorm and help the team to visualize ideas better than if they were working solo.

Communication will be at the heart of this project. As Artur Belka (2020) mentioned, in Agile Teams communication is improved because all team members belong to different area of expertise so that the decision-making process is decentralised. That means the product owner is not de only person who decides, it is a collaboration process, and each member makes informed decisions on its own area of knowledge. That eliminates the risk if the manager is unable to work or quit the. However, that is only possible because of team members are highly skilled and they foster excellent communication that keep the project progressing.

The Online Banking System will invest that agile communication approach as the figure below presents:



Discover 6 risk management strategies for software development | Digital product design and development company Boldare

Finally, this project will encourage leadership and negotiation skills to promote commitment among its members. The scrum master will ensure that team understand their roles, the cost and benefits and the reason behind the decisions and comprises the project will take. He will foster a culture of collaboration and listen to the team's opinion but taking assertive measures so that the project keeps progressing.

Task 9 – Project Methodology

The project will follow the Agile methodology, i.e. Scrum. That methodology suits this project better for many reasons. For instance, it has an iterative approach to deliver value to customers because it can implement small increments at a faster pace compared to other project management methodology, such as the Waterfall. It will often evaluate the projects plans, requirements, and results so that the development team can respond to changes rapidly.

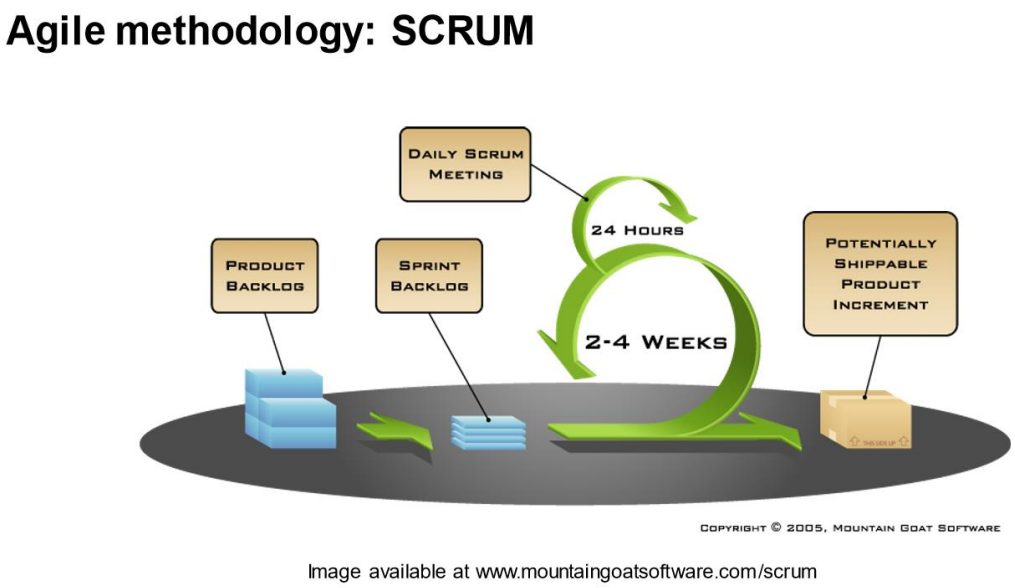
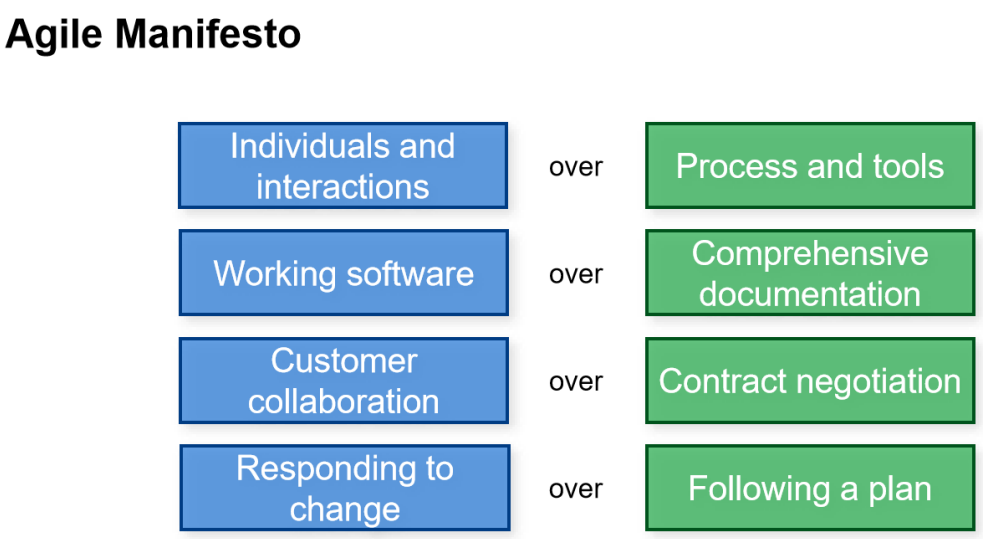


Image available at www.mountaingoatsoftware.com/scrum

The Online Book Publishing System project will be broken down into small tasks easy-to-manage chunks. So our customers will be able to see the result of the delivery of incremental changes on the Publishing System and they will be able to provide feedback for further improvements. That will improve communication within the whole team because it will set clear expectations of the project deliverables, enhancing a corporate culture of trust between customers and developers.

The project will focus on the 'Online Booking System' performance, it will react to changes quickly, emphasise collaboration with customers and knowledge sharing and interaction among team members. That is resembles the Agile manifesto as the image presents below:



Source: www.agilemanifesto.org

The Agile methodology is the most adequate for that type of IT project due to its peculiarities concerning the requirement of constant improvement to keep up with technological changes in the online book industry. For instance, to name a few of some small incremental improvements that this project foresees in different domains are described as follows:

Cybersecurity:

Login registration and revenue payment security will be tested and improvements and developers will include frequent system security improvements to keep. Therefore, the platform will be updated regularly to the most safe cybersecurity protocols available in the market. That will foster trust among the companies customers

GDPR and Data Protection policies:

The project will set regular reviews of the project agreements, i.e. the terms and conditions, to comply with the data protection regulations and policies and increase the company transparency on how we collect and process customers personal data. As data protection is a growing topic, not only in Europe, but world-wide, it is important to keep up-to-date with the GDPR.

Service Speed and reliability:

The project will review the platform coding to improve the publish system navigation such as a quick process to view, delete, update or upload books. The project will review processes to avoid interruption in the systems, e.g. books uploaded that have many files demanding a higher processing power to be uploaded will not 'crash' the platform. The system will ensure that they can be compressed to a certain format that it will not degrade the quality of the book.

User friendly design:

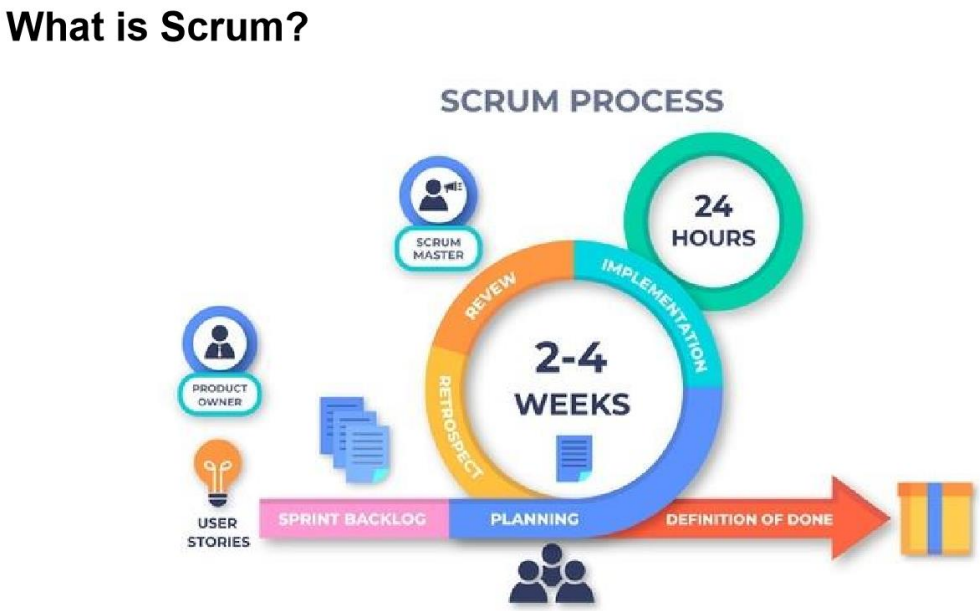
The project will frequently seek customers feedback on how to improve navigation within the platform. Customers will provide feedback about content they value and the developers will focus on making it accessible and easy to navigate. The quick-to-access design intends to help customers to provide a more intuitive experience that can save users time.

Improved services functionalities:

The project will improve the tools functionalities for new and existing books. For instance, the service will update the platform with new book formats, kindle, pdf, among others. Also, it will provide different types of fonts text value by customers, e.g. Amazon bookerly.

The rationale for the 'publishing online book' project to use Agile instead of other plan driven methodologies

The online Book Publishing System contains a medium size specialized team with wide experience in delivering projects using the Scrum software process model. That type of team tends to be more creative and strive in environments of freedom and chaos. The project will require the team to react very quickly to changes at any time of the project lifecycle. This project will follow the scrum approach as the figure below presents:



For instance, to keep the service competitive, the project will implement incremental changes, especially because new technologies are emerging in this field, such as cloud computing to provide real time updates on books uploaded, edited or deleted. Furthermore, it is likely that the technology will change very quickly in the near future and the project will be required to implement frequent updates to keep the service competitive. The fact that this consists of an IT project, the Agile methodology suits that because it is “effective for the dynamics development environment”.

Scrum vs. Other Models

	Software Process Model	Advantages	Disadvantages
Plan Driven	• Waterfall • Incremental Development • Iterative development • Spiral Development • Prototype Model • Rapid Application Development	• Suitable for large systems and teams. • Handles highly critical systems effectively. • Appropriate for stable development environment. • Require experienced personnel at the beginning. • Success achieved through structure and order.	• Longer length in each iteration or increment. • Cannot accommodate changes any time. • Lack of user involvement throughout the life cycle of the product. • Costly for the dynamic development environment. • Assume that, future changes will not occur.
Agile	• Scrum model • Extreme Programming (XP) • Dynamic System Development Method • Kanban • Feature Driven Development	• Suitable for small to medium systems and teams. • Can accommodate changes at any time. • Effective for the dynamic development environment. • Required expert agile personnel throughout the life cycle. • Success achieved through freedom and chaos.	• Not suitable for large systems (except FDD). • Shorter length in each iteration. • Can accommodate changes at any time. • Costly for the stable development environment. • Assume that, frequent future changes will occur.

More specifically, the 'Publish Booking Online System' will follow the Scrum approach due to the characteristics of this type of project management technique. For instance, it encourages a team responsive environment throughout the project; knowledge transfer; team creativity and flexibility during iterations. All of that is aligned with the Publish Booking Online System' and the choice for Scrum will increase the probability of project success.

This project will use the Scrum methodology because it has an Agile lightweight and simple process implementation. The Scrum methodology will allow to control and manage the software project development performing iterative and incremental changes to improve the product.

The project would have low probability of success if it adopts a plan driven software process model, for example the waterfall methodology, because they may follow a predetermined 'cookbook' approach that could limit team flexibility and creativity. The main reason for not choosing this approach was due to inability to quickly respond to changes in the market. That would be detrimental to an IT project such as the Booking Online System because the development team will need to react to its competitors and other market threats and implement new technologies at a fast pace. It would be too inefficient and costly to develop a long plan driven project and deliver a final product that no longer meets market requirements due to its fast pace changes.

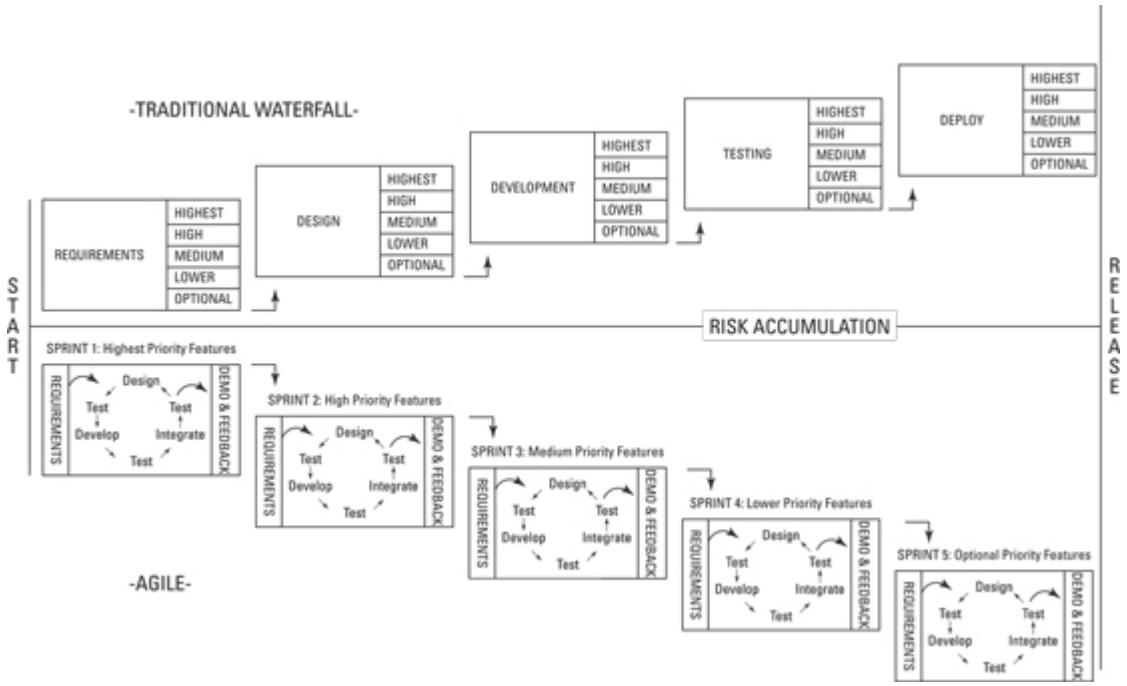
Therefore, the 'Booking Online System' preferred the Agile methodology, i.e. Scrum, instead of other 'driven plan', such as the Waterfall approach. That decision is based on all the evidence and arguments presented previously in this section.

Scrum vs. Other Models

Process Comparison

	Waterfall	Spiral	Iterative	SCRUM
Defined processes	Required	Required	Required	Planning & Closure only
Final product	Determined during planning	Determined during planning	Set during project	Set during project
Project cost	Determined during planning	Partially variable	Set during project	Set during project
Completion date	Determined during planning	Partially variable	Set during project	Set during project
Responsiveness to environment	Planning only	Planning primarily	At end of each iteration	**Throughout**
Team flexibility, creativity	Limited - cookbook approach	Limited - cookbook approach	Limited - cookbook approach	**Unlimited during iterations**
Knowledge transfer	Training prior to project	Training prior to project	Training prior to project	**Teamwork during project**
Probability of success	Low	Medium Low	Medium	**High**

Furthermore the Agile technique will not let risks accumulated and the project will be frequently tested for improvement at each iterations compared to the waterfall methodology:



Task 10 – Class Development

For class development, we chose to do the 'Register' use case from the use case diagram. The classes that would come into play when registering a new account are the author class (where it is presumed the registration details are stored), and the Unregistered Author class (where the function for registration is found).

Author Class

The class Author is where the data for the user details is stored, i.e. the email, first name and last name. The password can only be created when the user gets an email confirmation to confirm that their registration has been successful and is only included in the registered author class.

```
1  import java.util.Scanner;
2
3  public class Author {
4
5      public static void main(String[] args) {
6
7      String firstname;
8      String lastname;
9      String email;
10
11
12     }
13
14 }
15
```

Unregistered Author

The class Unregistered Author has no attributes. It only has an operator which is responsible for registering the user onto the system. It will achieve this by getting the attributes from the Author class, using them as input in the register() operator, and record these new entries.

```
1
2  public class UnregisteredAuthor {
3
4      public void register() {
5          // TODO - implement unregisteredAuthor.register
6
7      }
8
9  }
10
11
```

Registered Author

The registered author can login using the login() function where the attributes are retrieved using the user database from user input, validated and confirmed. From there the user can login into the system. Upon registration and email confirmation, the user will be asked to enter and confirm the new password they wish to set for their login. They would also have the option of resetting their password using a link sent to their registered email if they forget it.

```
1
2  public class RegisteredAuthor {
3
4      private String password;
5      private String publisherid;
6
7
8      public void login() {
9          // TODO - implement registeredAuthor.login
10
11     }
12
13     public void reset_password() {
14         // TODO - implement registeredAuthor.reset_password
15
16     }
17
18 }
19
20
```

Task 11 – Artefacts, Agile Methodology, User Stories, Backlogs and Burndown Charts

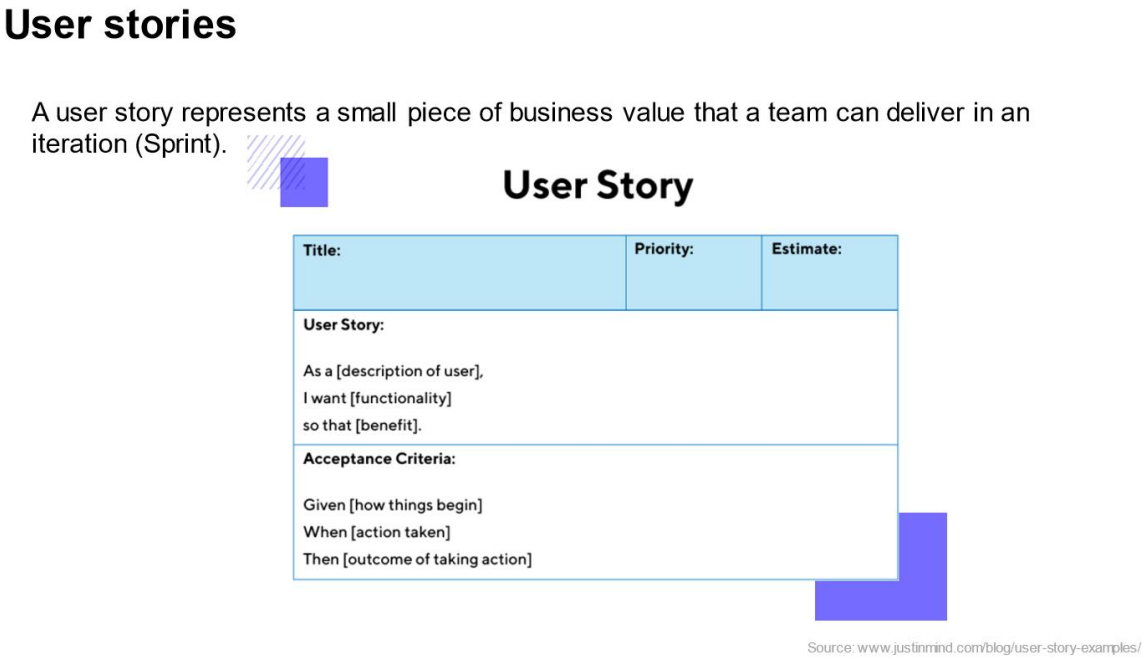
The Product backlog consists in a list of requirements, preferentially based on the 'user stories'.

User stories:

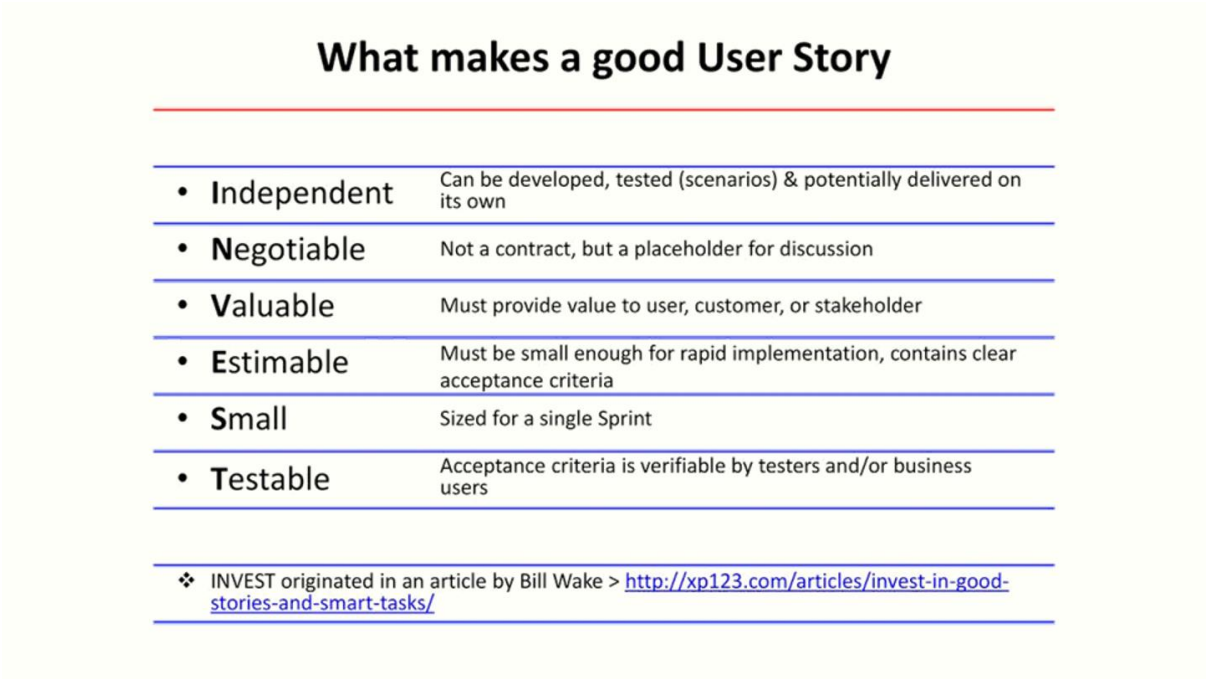
The project will use Agile methodology, so it will adopt 'user stories' instead of 'use cases'. It will follow the who, user role; what, the goal; and why, the reason to implement this story.

Those stories will approach a diversity type of shareholders, authors, publishers etc. with the A rating based on the effort to include those stories in 'Online Booking' functionalities.

The 'user stories' will need to add value to the business and be broken down into small manageable chunks, so that it can be implemented on a Sprint, i.e. iteration cycle. Please see below an example of the structure of an 'user story' :



The 'user stories' will follow the INVEST proposed on the article by Bill Wake. That means that they must be independent, i.e. test and delivered on its own; not a contract but a negotiated agreement; add value to the stakeholders; small enough for to be performed on a Sprint; and test its acceptance by users or business.



The 'Online Booking System' collected the following 'user stories':

1. As an author, I want to ensure that I have a fast and easy way to register and login into the Online Booking System, so that I can dedicate my time on other tasks, such as writing my books.
2. As a publisher, I want a friendly user interface and easy-to-use system that I can approve, reject, or suggest amendments on books, so that authors can take action on their proposed books quickly.
3. As an author, I want a reliable system with a version control that can automatically save all my edits so that I will not lose my edited work if case of an accident, i.e. a power cut.
4. As an author, I want a system that is reliable and will 'not crash' if I upload a 'heavy' file book with many images, so that I will not have to compromise the quality of my work.
5. As a publisher, I want the system to have an easy-to-understand interface to display approved books connected to the author name, so that I can release the payment to the correct author.
6. As an publisher, I want to have a secure method to pay authors, so that the company is not hacked and we can build trust among our stakeholders.
7. As an author, I want the system to let me choose a preferred payment method to receive for my published book, so that if I have any issue with one of my bank accounts I can quickly opt to receive my payment via Revolut or PayPal.

8. As an author, I want to be notified via a text message that my payment was processed, so that I don't have to check my account regularly.

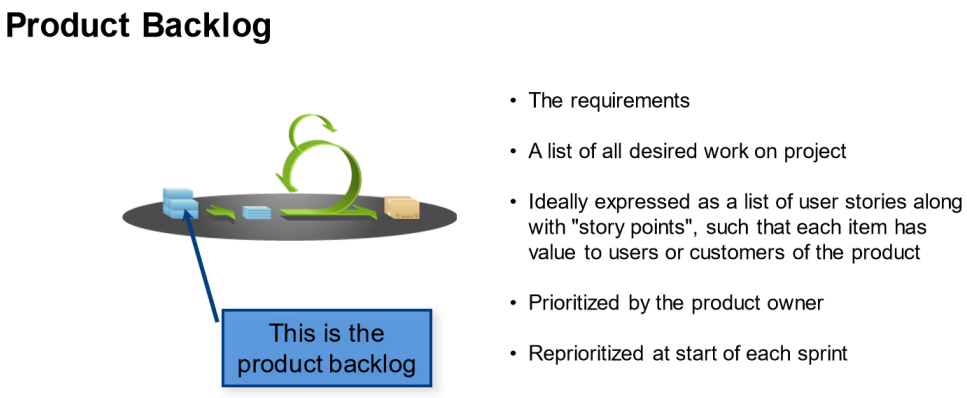
9. As a publisher, I want to have a system that allows me to make amendments quickly to the terms and conditions agreements, so that authors are informed if the company changes its policies.

10. As a publisher, I want a system that will be available 24h and will not crash, so that I can increase productivity and finish my work by the end of business hours, 6pm.

The Online Booking project backlog are:

Product backlog:

It will consist of all requirements of the project. It aims to accomplish all desired work the project intends to achieve. The product owner will prioritize it based on the list of user stories points, so that the product can add value to customers and stakeholders. At the start of each sprint, they will be reprioritized.



The product backlog will contain items based on the user stories (with an estimative of point):

High priority **Product Backlog for the Online Booking System**

Backlog Item	Estimate	Priority
Allow the system to have easy-to-use secured payment method	*5 (story points)*	*High*
As a publisher, I want the system to have an easy-to-understand interface to display approved books connected to the author's name, so that I can release the payment to the correct author.	5	*High*
As a publisher, I want to have a secure method to pay authors, so that the company is not hacked, and we can build trust among our stakeholders.	5	*High*
As an author, I want the system to let me choose a preferred payment method to receive for my published book, so that if I have any issue with one of my bank accounts I can quickly opt to receive my payment via Revolute or PayPal.	3	*Medium*
As an author, I want to be notified via a text message that my payment was processed, so that I do not have to check my account regularly.	2	*Low*
	20	***High***

Backlog Item	**Estimate**	**Priority**	
A reliable System	*5 (story points)*	*High*	
As an author, I want a reliable system with a version control that can automatically save all my edits so that I will not lose my edited work if case of an accident, i.e. a power cut	4	*Medium*	
As an author, I want a system that is reliable and will 'not crash' if I upload a 'heavy' file book with many images, so that I will not have to compromise the quality of my work.	5	*High*	
As a publisher, I want a system that will not crash, when using its main functionalities, e.g. approving books to be publish, so that I can increase productivity and finish my work by the end of business hours.	4	*Medium*	
	18	***Medium***	
Low Priority			

High priority				
System that is fast with a user-friendly interface	*5 (story points)*	*High*		
As a publisher, I want a friendly user interface and easy-to-use system that I can edit,(approve, reject, or suggest amendments on books), so that authors can take action on their proposed books quickly	5	*Medium*		
As an author, I want to ensure that I have a fast and easy way to register and login into the Online Booking System, so that I can dedicate my time on other tasks, such as writing my books.	3	*Medium*		
As a publisher, I want to have a system that allows me to make amendments quickly to the terms and conditions agreements, so that authors are informed if the company changes its policies.	3	*Medium*		
	16	***Medium***		

Low Priority

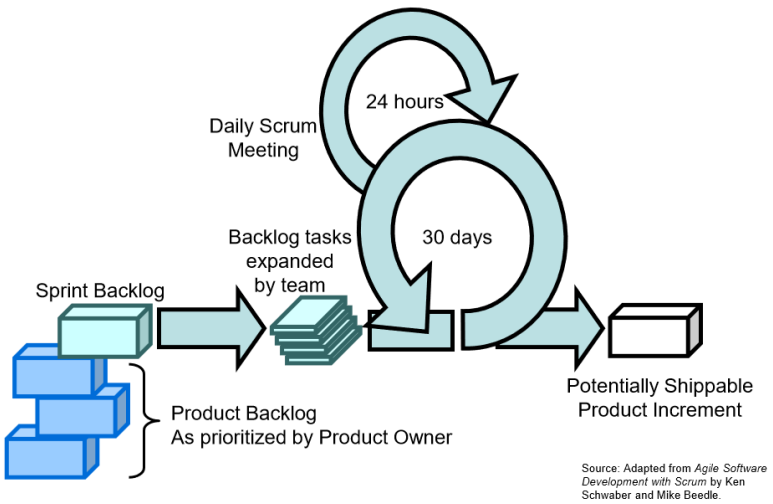
Sample Product Backlog

Backlog item	Estimate
Allow a guest to make a reservation	3 (story points)
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
...	30
...	50

Sprint backlog:

As previously mentioned, this project will focus on the Agile, Scrum process. The product owner will prioritize the product backlog, from where sprint backlogs will originate. The figure below pictures this process:

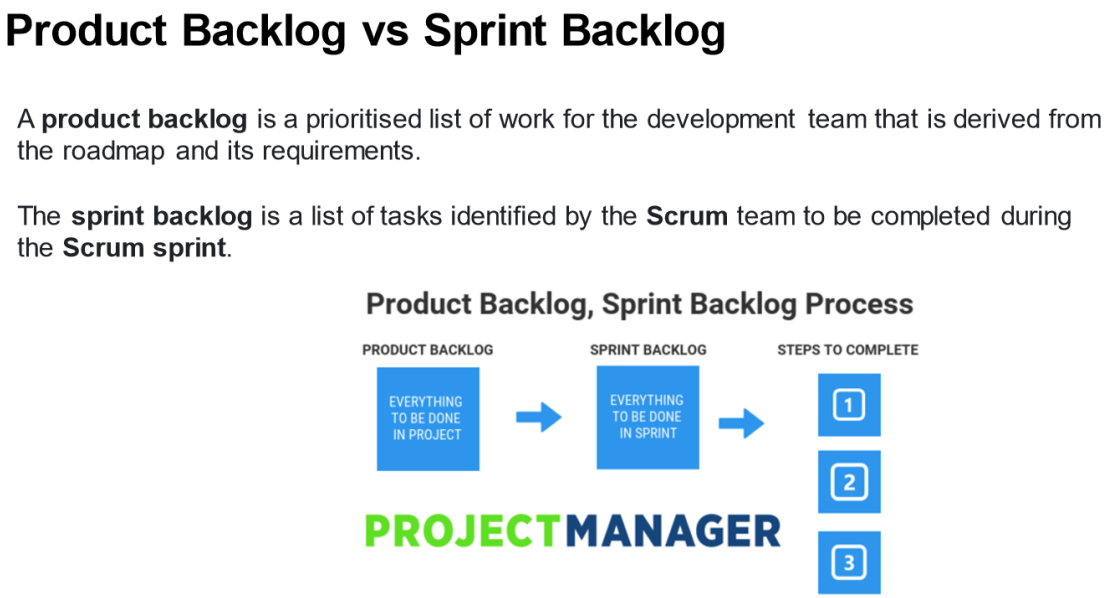
Scrum at a glance



Team members will choose the work they want to do from the sprint, i.e. tasks will never be assigned to any team member. The team will update their progress daily with the results achieved and the remaining work to be concluded.

Any team member will be able to make changes, e.g. edit, delete, on each other work from the sprint backlog, so that the work will be incrementally improved by the team. The sprint backlog will redefine and break down work items that still demand a large amount of time. That will apply for tasks that are not clearly defined.

This project will work on two different types of backlogs. For instance, the product backlog will focus on the overall product requirements. The Sprint Backlog will consist of smaller blocks of maneagle tasks from the product backlog. The image below demonstrate the main differences between the two backlogs this project will use:



Sprint planning:

There will be a team of ten people who will attend daily stand up meetings, 30 min maximum, to provide feedback, updates of their work, insights and knowledge sharing. The tasks from the sprint backlog will feed from product backlog and user stories items. The product backlog will be often evaluated so that it will create sprint goals. All sprint backlogs will estimate, in hours, the duration of tasks considering external factors such as the team capacity, the level of difficulty to implement the product back, business conditions, the technology available, among others . The image below presents how this project will plan its sprints backlogs:

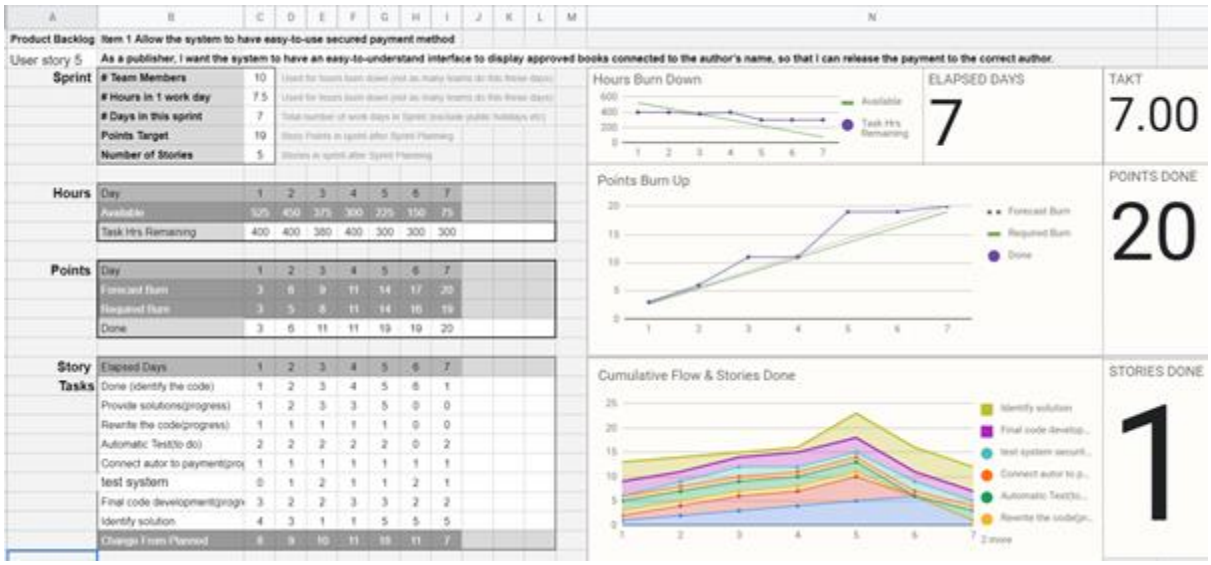
Sprint Execution:

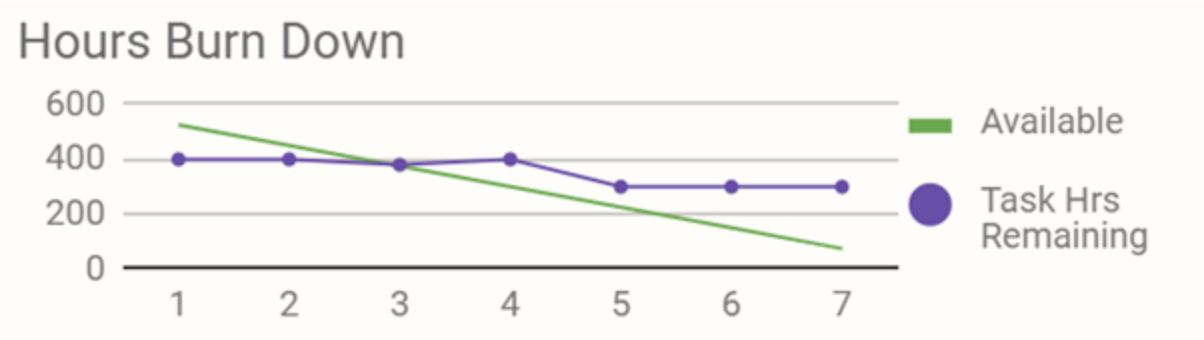
Communication

The project will follow the Sprint Backlog structure as the sample below:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Sprint 1												
2	02/03/2021												
3				Sprint day		1	2	3	4	5	6	7	
4						Monda	Tuesd	Wedı	Thurs	Frida	Satuı	Sunday	
5				Hours remaining		150	150	150	150	150	150	150	
6	Backlog Item:		1										
7	User story:	As a publisher, I want the system to have an easy-to-understand interface to display approved books connected to the author's name, so that I can release the payment to the correct author											
8	Tasks:		Owner	Estimate									
9	High	identify the code(Done)	Jacob	8		8	8	8	8	8	8	8	
10	High	Provide solutions(progress)	Tara	8		8	8	8	8	8	8	8	
11	High	Rewrite the code(progress)	Aurimas	8		8	8	8	8	8	8	8	
12	High	Automatic Test (in progress)	John	8		8	8	8	8	8	8	8	
13	High	Connect autor to payment(progress)	Mario	8		8	8	8	8	8	8	8	
14	High	test system security(progress)	Jane	10		10	10	10	10	10	10	10	
15	High	Final code development(progress)	Pedro	8		8	8	8	8	8	8	8	
16	High	Identify solution	Michael	8		8	8	8	8	8	8	8	
17													

Burndown Charts:




Hours Burn Down
600
400
200
0
1
2
3
4
5
6
7
Available
Task Hrs Remaining

Task 12 – Test Case Scenarios

These test case scenarios are meant to test the validation of the registration details, to see if each scenario would work correctly. For the test cases below, the user will need to input a new email (username in this case), password, first name and last name for identification purposes. For each test case, the test case template was used. Since it is too wide to be included in this document, screenshots of the left half and the right half separately. Since Java cannot store a user database in it's code, it would require a relational database like SQL to store user details, but using Java we can validate user inputs to see if these are valid inputs.

Email Validation:

SCREEN NAME	Registration Section
PREPARED BY	Aurimas Jurgutis
DESIGNATION	Email Input Testing
DATE	2021-05-05

SCENARIO ID	S1	SCENARIO DESCRIPTION	New Author Registration (new email t

S.NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT
S1	1	Valid new email entered	User prompted to enter email	Email input = john.johnson@email.com	You have entered: "john.johnson@email.com", email appears to be in correct format
S1	2	Invalid email entered: does not match correct format	User prompted to enter email	Email input = johnjohnsonemail.com	You have entered: "john.johnson@email.com", doesn't match the correct format
S1	3	Invalid email entered: email already matches existing email	User prompted to enter email	Email input = john.johnson@email.com	You have entered: "john.johnson@email.com", email already exists in our system
S1	4	Invalid email entered: input is blank	User prompted to enter email	Email input = "" (blank)	You have entered: "", input is blank, please input an email address

testing)

POSTCONDITION	ACTUAL RESULT	STATUS	DEFECT ID	COMMENTS
User presses "Enter" on keyboard to input new email	You have entered: "john.johnson@email.com", email appears to be in correct format	Pass		
User presses "Enter" to go back to the email input to try again	You have entered: "john.johnson@email.com", doesn't match the correct format	Pass		
User presses "Enter" to go back to the email input to try again	You have entered: "john.johnson@email.com", email already exists in our system	Pass		
User presses "Enter" to go back to the email input to try again	You have entered: "", input is blank, please input an email address	Pass		

Password Validation:

SCREEN NAME		Registration Section	
PREPARED BY		Aurimas Jurgutis	
DESIGNATION		Password Input Testing	
DATE		2021-05-05	
SCENARIO ID	S2	SCENARIO DESCRIPTION	New Author Registration (new passwor

S.NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT
S2	1	Valid new password entered	User prompted to enter password	Password input = johnson123	You have entered: "johnson123", password appears to be correct
S2	2	Invalid password entered: does not match requirements	User prompted to enter password	Password input = johnson\][\	You have entered: " johnson\][\", password doesn't match the correct format
S2	3	Invalid password entered:password is too short	User prompted to enter password	Password input = johnson	You have entered: "johnson", password is too short
S2	4	Invalid password entered: input is blank	User prompted to enter password	Password input = "" (blank)	You have entered: "", input is blank, please input a password

d testing)

POSTCONDITION	ACTUAL RESULT	STATUS	DEFECT ID	COMMENTS
User presses "Enter" on keyboard to input new password	You have entered: "johnson123", password appears to be correct	Pass		
User presses "Enter" to go back to the password input to try again	You have entered: " johnson\][\", password doesn't match the correct format	Pass		
User presses "Enter" to go back to the password input to try again	You have entered: "johnson", password is too short	Pass		
User presses "Enter" to go back to the password input to try again	You have entered: "", input is blank, please input a password	Pass		

First Name Validation:

SCREEN NAME	Registration Section
PREPARED BY	Aurimas Jurgutis
DESIGNATION	Author's First Name Input Testing
DATE	2021-05-05

SCENARIO ID	S3	SCENARIO DESCRIPTION	New Author Registration (author name

S.NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT
S3	1	Valid first name entered	User prompted to enter first name	Name input = John	You have entered: "John", name appears to be correct
S3	2	Invalid name entered: name input must include letters only	User prompted to enter first name	Name input = John';;[	You have entered: "John';;[", name doesn't match the correct format
S3	3	Invalid name entered: name is too short (must be at least 4 characters long)	User prompted to enter first name	Name input = Joh	You have entered: "Joh", name is too short
S3	4	Invalid name entered: input is blank	User prompted to enter first name	Name input = "" (blank)	You have entered: "", input is blank, please input a name

e testing)

POSTCONDITION	ACTUAL RESULT	STATUS	DEFECT ID	COMMENTS
User presses "Enter" on keyboard to input new name	You have entered: "John", name appears to be correct	Pass		
User presses "Enter" to go back to the name input to try again	You have entered: "John';;[", name doesn't match the correct format	Pass		
User presses "Enter" to go back to the name input to try again	You have entered: "Joh", name is too short	Pass		
User presses "Enter" to go back to the name input to try again	You have entered: "", input is blank, please input a name	Pass		

Last Name (Surname) Validation:

SCREEN NAME	Registration Section	
PREPARED BY	Aurimas Jurgutis	
DESIGNATION	Author's Last Name Input Testing	
DATE	2021-05-05	
SCENARIO ID	S4	SCENARIO DESCRIPTION	New Author Registration (author surname i

S.NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT
S4	1	Valid last name entered	User prompted to enter surname	Surname input = Johnson	You have entered: "Johnson", surname appears to be correct
S4	2	Invalid surname entered: name input must include letters only	User prompted to enter surname	Surname input = Johnson...,,;	You have entered: " johnson...,,;", surname doesn't match the correct format
S4	3	Invalid surname entered: name is too short (must be at least 4 characters long)	User prompted to enter surname	Surname input = j	You have entered: "j", surname is too short, should be at least 2 characters long
S4	4	Invalid surname entered: input is blank	User prompted to enter surname	Surname input = "" (blank)	You have entered: "", input is blank, please input a surname

input testing)

POSTCONDITION	ACTUAL RESULT	STATUS	DEFECT ID	COMMENTS
User presses "Enter" on keyboard to input new name	You have entered: "Johnson", surname appears to be correct	Pass		
User presses "Enter" to go back to the name input to try again	You have entered: " johnson...,,;", surname doesn't match the correct format	Pass		
User presses "Enter" to go back to the name input to try again	You have entered: "j", surname is too short, should be at least 2 characters long	Pass		
User presses "Enter" to go back to the name input to try again	You have entered: "", input is blank, please input a surname	Pass		

Confirm Password Validation:

SCREEN NAME	Registration Section		
PREPARED BY	Aurimas Jurgutis		
DESIGNATION	Author's Password Confirmation Input Testing		
DATE	2021-05-05		
SCENARIO ID	S5	SCENARIO DESCRIPTION	New Author Registration (input new password and c

S.NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT
S5	1	Inputs for password and confirm new password match	User prompted to enter password	Password inputs = johnson123, johnson123	Congratulations, you have successfully created a new account!
S5	2	Inputs for password and confirm new password do not match	User prompted to confirm new password	Password inputs = johnson123, johnson	Passwords do not match, please try again.

onfirm new password)

POSTCONDITION	ACTUAL RESULT	STATUS	DEFECT ID	COMMENTS
User presses "Enter" to proceed to login	Congratulations, you have successfully created a new account!	Pass		
User presses "Enter" to go back to the password creation UI	Passwords do not match, please try again.	Pass		

Conclusions

The Online Booking System project will use the Agile methodology, instead of other plan driven methodology, e.g. the Waterfall processes. That will help to address wide range of internal and external risks because each deliverable will perform several iterations via sprint backlogs considering stakeholders feedback for product improvement. That will also help to ensure quality is at a high standard.

The project will use Agile to foster good communication to ensure that all team members collaborate with insights and foster knowledge sharing. It will use technology to improve performance and communication and limit daily stand-up meeting team at a maximum of 30 minutes.

The project will implement use case stories to capture ideas and functionalities that are most valuable to customers. There will feed the 'Online Booking System' product backlog and the sprint backlogs. Burn charts will contain the graph to display the progress of the work remaining. All functionalities will be exhaustively tested via the test case temple to ensure the software is working properly.

References

- 1. Paul K. Gellert, 2015. Optimism and Education: The New Ideology.
- 2. Rothstein, Huber and Gaskell, 2006: A theory of risk colonization .
- 3. Peter Harremoes, 2013: Mutual Information of Contingency Tables.
- 4. Black and Baldwin, 2010: Really Responsive Risk-Based Regulation.
- 5. Artur Belka, 2020: Customer Success Guide, Boldare.