Software Engineering Project Management

Team 1 **Project Report**

18 April 2022

Team Members

Name	ePortfolio
Andrey Smirnov	https://soundmaven.github.io/e-portfolio/
Michael Justus	https://micjustus.github.io/essex-eport2/
Taylor Edgell	https://tedgell.github.io/MSc-ComputerScience/
Grace Clarke	https://gclarke95.github.io/University/
Shan Swanlow	https://shans96.github.io/

Document Version History

Version	Date	Author	Details
0.1	22/03/22	Grace Clarke	Document creation
0.2	27/03/22	Taylor Edgell	Heading formatting
0.3	29/03/22	Andrey Smirnov	TOGAF section
0.4	29/03/22	Michael Justus	Review
0.5	30/03/22	Andrey Smirnov	Edits and Appendix entries
0.6	31/03/22	Taylor Edgell	Gherkin statements
0.7	03/04/22	Andrey Smirnov	Scrum methodology
0.8	03/04/22	Shan Swanlow	User stories, Gherkin statements
0.9	04/04/22	Michael Justus	SMART requirements. Review, content update
0.10	04/04/22	Taylor Edgell	Initial requirements, Design description, and refactoring
0.11	05/04/22	Michael Justus	Update smart requirements
0.12	06/04/22	Grace Clarke	SMART methodology
0.13	10/04/22	Michael Justus	Review
0.14	13/04/22	Andrey Smirnov	U.I. wireframes
0.15	14/04/22	Taylor Edgell	Use Case Diagram, Demo, and timeline justifications

0.16	15/04/22	Grace Clarke	Updates & Reviewing
0.17	15/04/22	Shan Swanlow	Updated Gherkin statements
0.18	15/04/22	Michael Justus	Revision
0.19	17/04/22	Taylor Edgell	Revision
0.20	18/04/22	Andrey Smirnov	Revision
0.21	18/04/22	Grace Clarke	Final Review
1	18/04/22	Team	Finalisation

Definitions and Abbreviations

Acronym	Description
BDD	Behavioural Driven Development
SCRUM	Agile software development methodology
SMART	Specific, Measurable, Actionable, Realistic, Timely
TOGAF	The Open Group Architecture Framework
U.I.	User interface
USP	Unique selling point

Word Count (Excluding titles and captions): 1076

Contents

1. Design Description

1.1 Overview

The product design is for an educational “Flip card” memory video game based on customer feedback targeted at mobile devices running on the Android operating system and with a child-friendly design. A vital requirement for the product is to be educational because “there is a general agreement among educators that games [positively affect] achievement, problem-solving, …, and engagement in task learning” (da Silva et al., 2019).

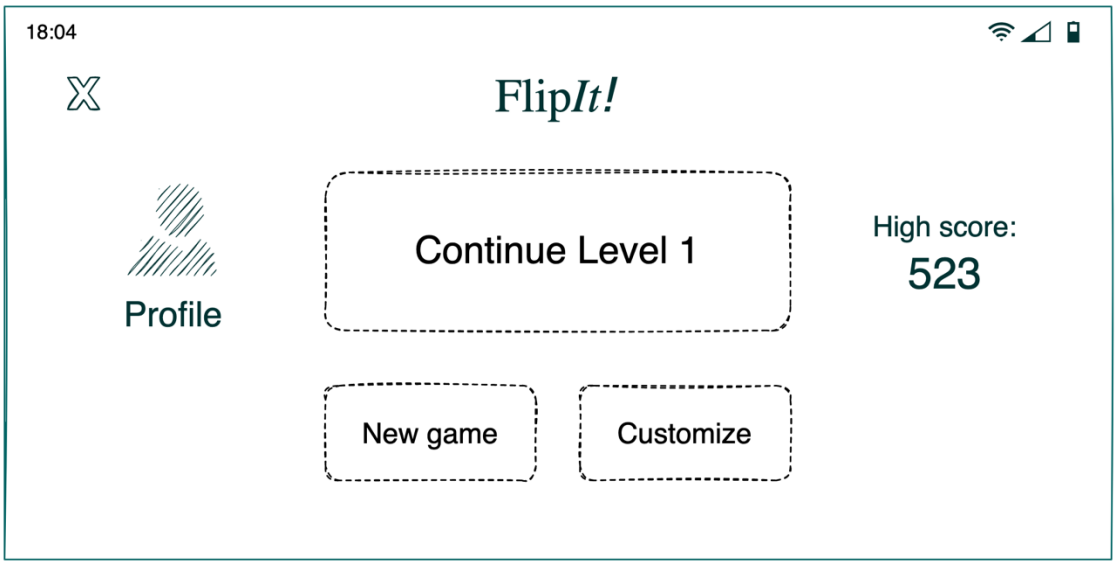

Figure 1 - Wireframe - Start Screen

The “Flip card” mechanic is a classic video game archetype that requires matching key images to key words. Level progression is linked to each user’s profile to allow progress to be saved and restored.

Difficulty in progressive levels is managed through variable time limits, and question difficulty.

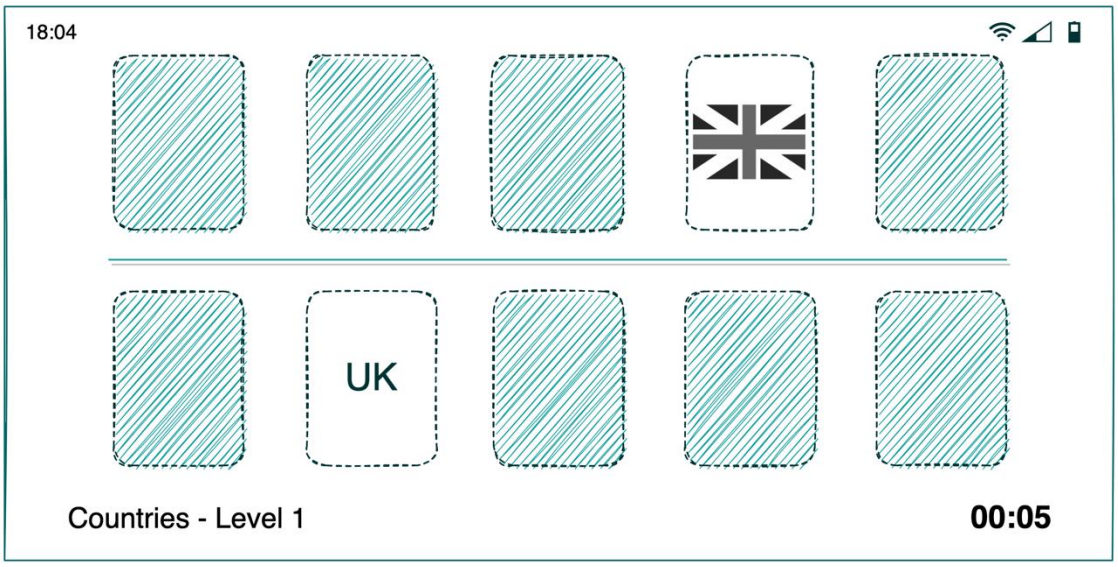

Figure 2 - Wireframe - Level design

2. Methodology

2.1 Scrum

2.1.1 Overview

Our chosen product methodology is Scrum. A full explanation of Scrum can be seen within appendix 6.3.

2.1.2 Benefits and Considerations

The adoption of Scrum has a positive impact on software quality, development costs and time required to complete a project (Hayat et al., 2019). One consideration for choosing Scrum is that Scrum is fundamentally an *iterative* model and does not have the notion of a fixed deadline. While practitioners have proposed solutions that circumvent this limitation (Hatrick-Smith, 2020), it can be argued that these attempts are contradictory to the core principles of the Scrum framework. Additionally, due to constraints with online study, it is impossible to regularly execute certain essential Scrum activities such as Daily Standups and Sprint Reviews.

2.1.3 Justification

The team chose Scrum because:

- It prioritises working software over extensive documentation and avoids elaborate, protracted planning associated with traditional “Waterfall-like” methodologies (Sharma & Hasteer, 2016).
- According to Lei et al. (2017), Scrum works best in small cross-functional teams and supports generalists and specialists, matching our team’s size and composition.

Also considered was Kanban, an alternate agile software development methodology with similar benefits to Scrum (Ahmad et al., 2018). Ultimately, Scrum is preferred because of its focus on time-boxed iterations and work increments, which aligns well with the structure of the SEPM module.

2.2 TOGAF

To improve operational efficiency during our product development we have considered applying concepts of The Open Group Architecture Framework (“TOGAF”) towards our project. Further information can be seen from appendix 6.6.

2.3 SMART

Throughout the project we have considered SMART to help refine and improve user requirements. More information on SMART can be seen in appendix 6.3.

3. Requirements

3.1 Initial Requirements

The initial requirements were gathered through discussions with the customer from a pre-populated list of requirements. The initial accepted requirements shown in 6.1.

3.2 Final Requirements (SMART)

Mapping the initial requirements to SMART requirements allowed the team to consider each customer requirement in-depth to facilitate more structured development. The final requirements can be seen below:

Table 1 Customer requirements restated as SMART requirements

Refined Requirement #	Requirement	Original Requirement #
1	*A user can play the game with either their left or right hand.*	*2*
2	*The game will be updated via an internet connection to regularly update score data after level completion and when loading a new gaming session.*	*8*
3	*A child can use their fingers to press on the device's screen to initiate an action or respond to a visual prompt.*	*18*
4	*The game keeps children engaged by providing levels of increasing difficulty.*	*22*
5	*The game presents facts about nature, people, or places. The information is relevant based on the child's profile age.*	*23*
6	*The game allows the child's parent or guardian to enter their child's details. The profile will only collect a username and will track game progress.*	*24*
7	*The game is played by children aged 6 and 12. All textual content will be age appropriate. The game will prioritise pictures and animations over text.*	*26*
8	*Data captured and persisted by the toy will not exceed 1MB. The format for data storage will be in a human-*	*28*

	readable format. Data stored includes interactions, level-up details and progress.	
9	*The toy allows parents to view their child's progression history via a menu. The data is presented in a table format and displays a default collection of 10 rows. Each table of information is paginated and allows the parent the sort of each column.*	*29*
10	*USP is the toy is gender neutral.*	*30*

To meet the above requirements, the game must support specific behaviours and menus which are shown in the form of a UML use case diagram (Figure 3).

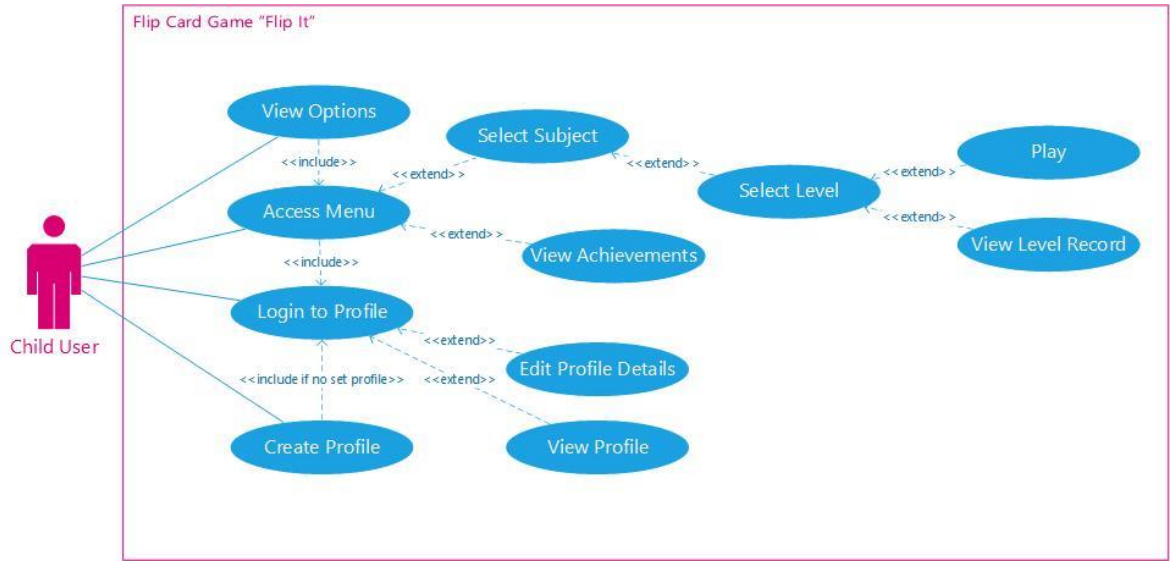

Figure 3 Use cases for system

3.3 User Stories

Having obtained customer requirements, it is necessary to adapt them to better fit the Scrum methodology by narrowing the scope around each requirement. We therefore map the requirements to user stories: in Scrum methodology, they populate a product backlog for development sprints (Tsilionis et al., 2021; Wautelet

et al., 2019). The SMART requirements were mapped to user stories to drive development tasks and sprint planning (user stories are found in Appendix A).

3.4 Gherkin

As a final step in preparing our requirements to fit the Scrum methodology, we create Gherkin statements as shown in 6.7.

4. Project Plan

4.1 Sprint Management/Timeline

The sprints for deliverables will be managed using the cloud-based product management software ClickUp. Each deliverable is captured as a single task and allocated to relevant development team members. The proposed project timeline is shown in Figure 4. It should be noted the Gantt chart concept does not formally align to Scrum, but was included upon customer request.

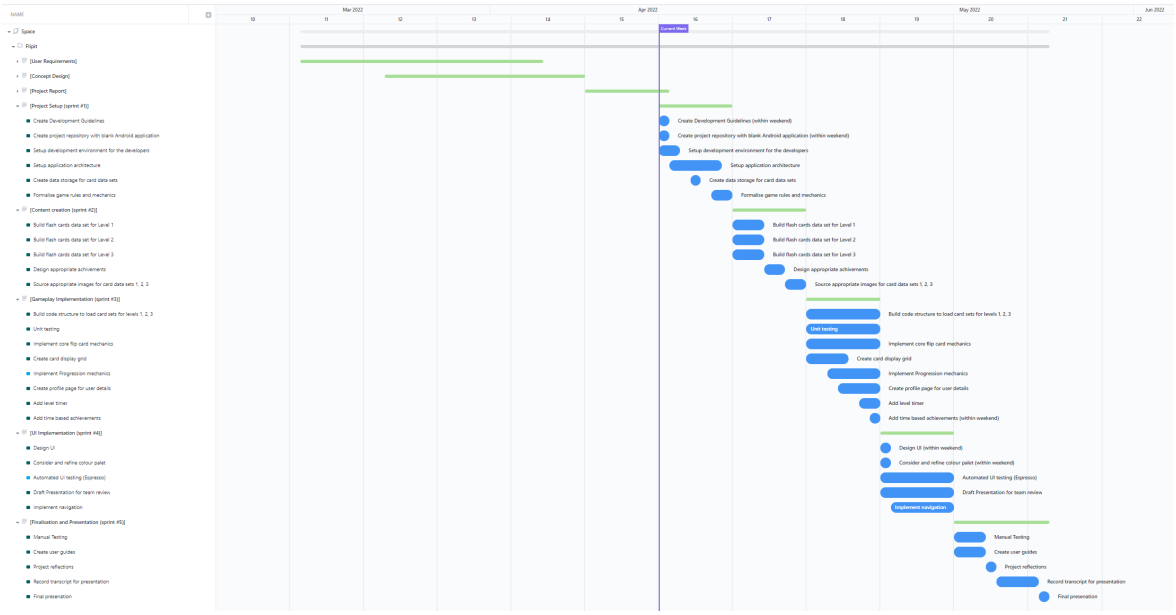

Figure 4 Project Gantt Chart

4.2 Implementation Estimations

The development timeline was created using expert-based judgment, one of the most common estimation techniques for Scrum (Jorge-Martinez et al., 2021).

Each of the ten developed user stories (6.2 User Stories Table) was discussed during a team planning meeting, and appropriate story points were assigned in ClickUp to the tasks. This was done practically through planning poker.

In determining delivery estimates, the team was mindful of its size and time constraints, as well as of the fact that using story points for effort estimation might lead to inaccurate estimates (Zahraoui & Idrissi, 2015).

4.3 Demo

The demo will provide a “Lite” version of the full flip card game that partially demonstrates specific requirements. The list of committed demo functionality can be seen in Appendix 6.3.

We will consider the project to be successful if we are to create a demo with all the promised functionalities.

Word Count (Including titles and captions): 1076

5. References

Ahmad, M., Ovais, D., Conboy, K. & Oivo, M. (2018) Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software* 137(1): 96-113.

Alm, R. & Wibotzki, M. (2013) 'TOGAF adaption for small and medium enterprises', *International Conference on Business Information Systems*. Poznan, 25-27 April. Berlin: Springer. 112-123.

da Silva Júnior, J., Uchoa, D., Sousa Lima, M. & Monteiro, A. (2019) Stereochemistry game: Creating and playing a fun board game to engage students in reviewing stereochemistry concepts. *Journal of chemical education*: 96(8): 1680-1685.

dos Santos, E. & Vilain, P. (2018) 'Automated Acceptance Tests as Software Requirements: An Experiment to Compare the Applicability of Fit Tables and Gherkin Language', *Agile Processes in Software Engineering and Extreme Programming (XP 2018).* Porto, Portugal, 21-25 May. Cham: Springer. 104-119.

Esposito, F., Wang, J., Contoli, C., Davoli, G., Cerroni, W. and Callegati, F. (2018). A behavior-driven approach to intent specification for software-defined infrastructure management. *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*: Verona, 27-29 November. New York: IEEE. 1-6.

Hatrick-Smith, E. (2020) Practical Scrum Delivery within a Project Deadline. Available from: https://medium.com/nib-tech-travel/practical-scrum-delivery-within-a-project-deadline-e67d842bac34 [Accessed 2 April 2022].

Hayat, F., Rehman, A., Arif, K., Wahab, K. & Abbas, M. (2019) 'The Influence of Agile Methodology (Scrum) on Software Project Management', *20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. Toyama, 8-11 July. New York: IEEE. 145-149.

Jorge-Martinez, D., Misra, S., Butt, S. & Ayeni, F. (2021) 'Estimation Techniques for Scrum: A Qualitative Systematic Study', *12th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2021)*. Online, 16-18 December. Cham: Springer. 818-829.

Lei, H., Ganjeizadeh, F., Jayachandran, P. & Ozcan, P. (2017) A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer Integrated Manufacturing* 43(C): 59-67.

Mannion, M. and Keepence, B., 1995. SMART requirements. *ACM SIGSOFT Software Engineering Notes*, 20(2): 42-47.

Minichiello, A., Hood, J.R. and Harkness, D.S., 2018. Bringing user experience design to bear on STEM education: A narrative literature review. *Journal for STEM Education Research*: 1(1): 7-33.

Kornyshova, E. & Barrios, J. (2021) Process-oriented Knowledge Representation of the Requirement Management Phase of TOGAF-ADM: an Empirical Evaluation. *Procedia Computer Science* 192(1): 2239-2248.

Kotusev, S. (2016) The critical scrutiny of TOGAF. Available from: https://www.bcs.org/articles-opinion-and-research/the-critical-scrutiny-of-togaf/ [Accessed 29 March 2022].

Reselman, B. (2020) TOGAF and the history of Enterprise Architecture. Available from: https://www.redhat.com/architect/togaf [Accessed 29 March 2022].

Scrum.org (N.D.) What is Scrum? Available from: https://www.scrum.org/resources/what-is-scrum [Accessed 31 March 2022].

Sharma, S. & Hasteer, N. (2016) 'A comprehensive study on state of Scrum development', *2016 International Conference on Computing, Communication and Automation (ICCCA)*. Greater Noida, 29-30 April. New York: IEEE. 867-872.

Schwaber, K. & Sutherland, J. (2012) *Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, and Leave Competitors in the Dust*. 1st ed. Hoboken: Wiley.

The Open Group (2001) Business Scenarios. Available from: www.opengroup.org/architecture/togaf7-doc/arch/p4/bus_scen/bus_scen.htm [Accessed 29 March 2022].

The Open Group (2001) Introduction to the Architecture Development Method (ADM). Available from: www.opengroup.org/public/arch/p2/p2_intro.htm [Accessed 29 March 2022].

Tsilionis, K., Maene, J., Heng, S., Wautelet, Y. & Poelmans, S (2021) 'Conceptual Modeling Versus User Story Mapping: Which is the Best Approach to Agile Requirements Engineering?', *Research Challenges in Information Science* (*RCIS 2021)*. Limassol, Cyprus, 11-14 May. Cham: Springer. 356-373.

Wautelet, Y., Gielis, D., Poelmans, S. & Heng, S. (2019) 'Evaluating the Impact of User Stories Quality on the Ability to Understand and Structure Requirements', *IFIP Working Conference on The Practice of Enterprise Modeling* (*PoEM 2019)*. Luxembourg, 27-29 November. Cham: Springer. 3-19.

Zahraoui, H. & Idrissi, M. (2015) 'Adjusting story points calculation in scrum effort & time estimation', *2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA)*. Rabat, 20-21 October. New York: IEEE. 1-8.

6. Appendix

6.1 Initial Requirements

Table 2 List of customer requirements

Requirement #	Requirement
2	The U.I. should be usable with 1 hand.
8	The device should be immune from internet attacks or include protection (e.g., virus scanner, firewall) as part of the system software.
18	The device should be controllable via touch, voice, or text/keyboard input.
22	Angela is a 35-year-old mum, and she wants the game to keep her son busy
23	Glenda is Adam's grandmother. She wants the game to be educational.
24	A player must be able to create a user profile.
26	Kashif is a 30-year-old dad. He wants the game to be safe.
28	Data must be stored in the most efficient manner possible.
29	Data must be searchable and managed as efficiently as possible.
30	Andrew works in a toy shop, and he wants the game to have a unique selling point (USP).

6.2 User Stories Table

As an <Actor>, I/we want <Goal or Objective> so that <Benefit, Result or Reason>

Table 3 Customer requirements

Story No.	Actor	Goal or Objective	Benefit, Result or Reason
1	Child User	Use the device with a single hand	The game can easily be played on a phone interface
2	Parent	Ensure the device is protected from malicious attack	The game can be played safely without worry about data security

3	Child	Be able to interact with the game through an interface	I can make decisions/ choices and engage with the game
4	Parent	Have the game be engaging to a child	It can keep them occupied while I do other activities
5	Guardian	Have the game provide educational benefit	The child learns from the game and gains beneficial information from it
6	Child	Be able to have a user profile	The game information is linked to me and follows my progress
7	Parent	Make sure the game is safe	The child is not given access to inappropriate information
8	Developer	The data is to be stored in the most efficient way	The game does not take up unnecessary space and becomes bloated and troublesome on the installed device.
9	Developer	Ensure that data must be able to be searched and managed as efficiently as possible.	The game is not slowed by data overhead and can perform optimally from a technical standpoint
10	Seller	Provide a game with a USP	To entices parents/children to want to purchase the game

6.3 Demo Functionality

Table 4 Demo functionality mapped to requirements

Requirement #	Demo functionality

1	*A mobile (Android) demo will be developed and to be usable with a simple control scheme*
2	*The developed demo will consider relevant vulnerabilities, protections, and good coding practises (e.g clean commenting) within the code.*
3	*The game demo will be accessible and controllable through a mobile interface*
4	*A set of three subject levels will be playable in the demo :* • *Country Flags* • *Landmarks* • *Dinosaurs*
5	*The demo will contain a child-friendly U.I. and display child-appropriate educational information.*
6	*The demo contains the functionality to create and view user profiles. User profiles will record game progress, and achievements received*
7	*Within the demo, all information will be age-appropriate and vetted before inclusion.*
8	*The demo version is lightweight and will be of indicative size in relation to the actual full game.*
9	*Parents can view the child's progress through the user profile section within the game demo.*
10	*Gender-neutral educational subjects have been chosen for the demo*

6.4 SCRUM

Scrum is an Agile framework that accommodates frequent changes in business requirements, reducing risks inherent in the software development process (Scrum.org, N.D.). Scrum is based on a relatively light, iterative, incremental core model, with a small team of people as a fundamental unit, known as a Scrum Team. This model, or empirical process as described by Schwaber and Sutherland (2012), is shown below:

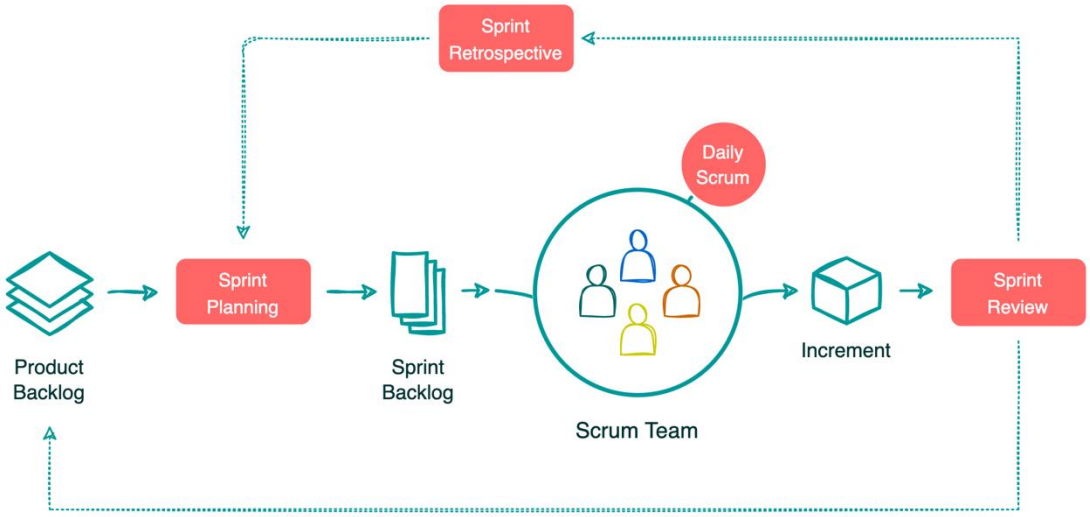

Figure 5 The Scrum process (Schwaber & Sutherland, 2012)

The Scrum process starts with a Product Vision defined by a Product Owner, who determines the contents of the Product Backlog. In Sprint Planning, the team decides which tasks to complete in the next sprint cycle. During the sprint, the group uses Daily Standups to discuss the direction and progress. At the end of each sprint, Sprint Reviews offer stakeholders an opportunity to provide feedback on the current product iteration. Finally, Sprint Retrospectives are conducted before the next Sprint Planning and are focused on overall workflow enhancements.

6.5 SMART

Ongoing discussions were had with the customer to liaise and agree on the requirements for the game design. The accepted requirements were then developed using the SMART methodology, making requirements easier to interpret, implement and track to completion (Mannion and Keepence, 1995).

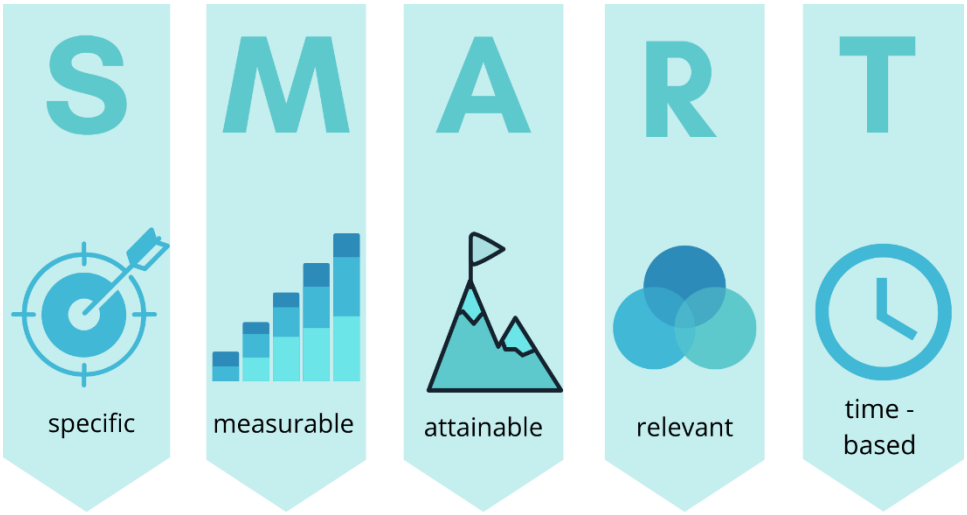

Figure 6 SMART definition

Utilising the SMART methodology means every requirement is verified as correct if it meets all the above, e.g., being specific, measurable, attainable, relevant and time-based.

6.6 TOGAF

The Open Group Architecture Framework ("TOGAF") is a famous enterprise architecture standard used to improve operational efficiency in large-scale companies (Kornyshova & Barrios, 2021). Although TOGAF's internal consistency has been questioned (Alm & Wibotzki, 2013; Kotusev, 2016), the following concepts are relevant from the project management perspective:

1. TOGAF provides a formal method of deriving technical characteristics from high-level business scenarios. The related conceptual model (Appendix 6.8) was consulted when analysing the scope of individual requirements and their potential technical impact.
2. Requirements management is central to all TOGAF Architecture Development Method (ADM) stages. This emphasises that requirements are not static and are subject to change (Appendix 6.9).

6.7 Gherkin

Although user stories are written in plain English, ambiguity may still arise during their development. Santos & Vilain (2018) therefore encourage using Gherkin statements for user stories because they are unambiguous, usable by non-technical people and verification-agnostic (Esposito et al., 2018). Figure 7

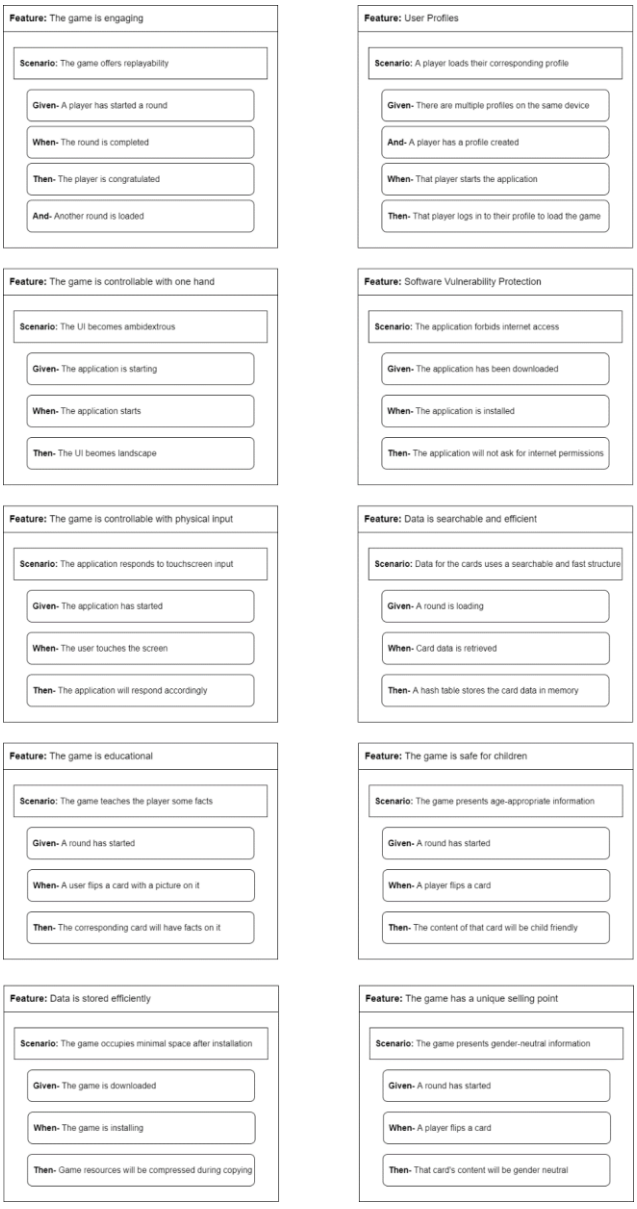

Figure 7 Gherkin statements for customer requirements

6.8 TOGAF Business Scenarios

According to The Open Group (2001), Business Scenarios help identify the needs of the business and gain “buy-in” from the key stakeholders involved in a software development project. The overall process of creating a Business Scenario can be seen below:

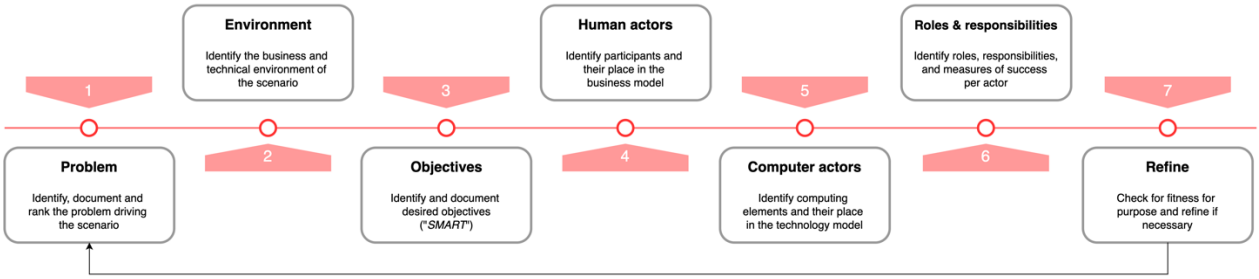

Figure 8 TOGAF Business Scenario creation process

According to the TOGAF 7 specification, a good Business Scenario needs to be SMART, a criterion already referred to in this document.

6.9 TOGAF Architecture Development Method

The TOGAF Architecture Development Method (ADM) is a generic architectural development method designed to support most system and organisational requirements that I.T. architects might be dealing with (Reselman, 2020).

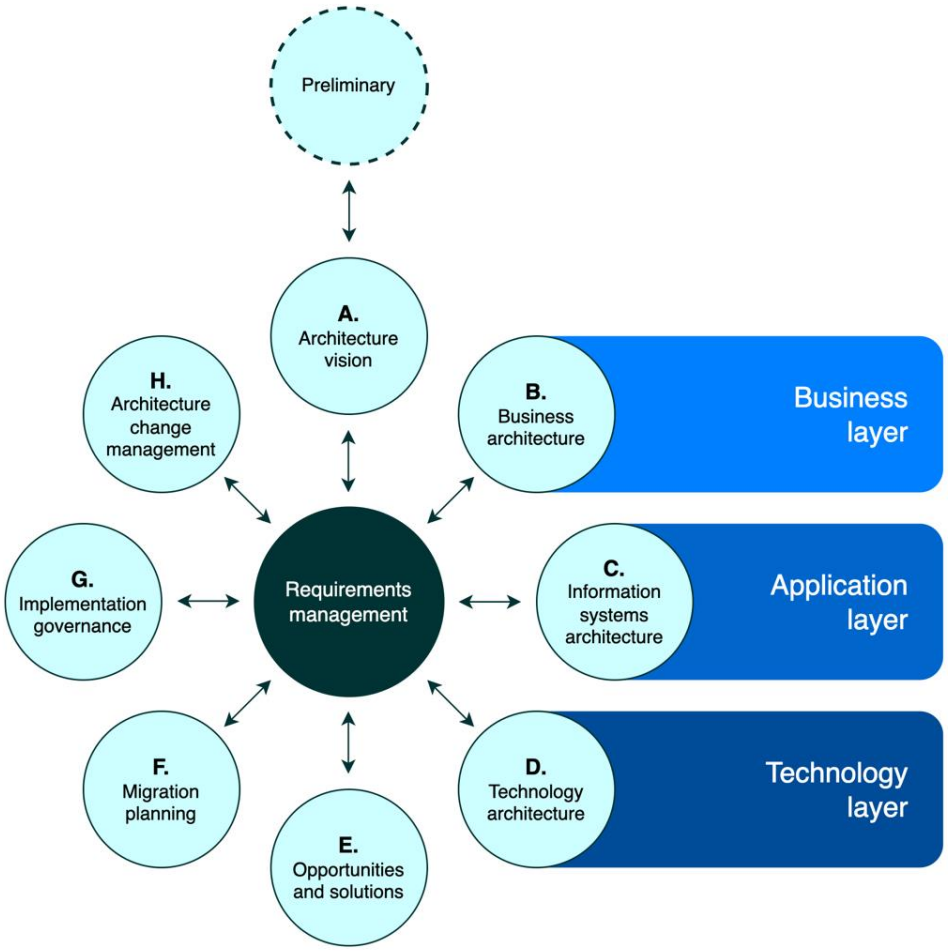

Figure 8 TOGAF Architecture Development Method

While the ADM is not directly applied in our Development Team Project, the conceptual model above provided insight into the link between business requirements and various enterprise architecture processes, such as developing an architecture vision or implementation governance.

Interestingly, the ADM documentation mentions that the preferred way of architecture development is iterative and cyclical and should incorporate frequent validation of results against the original business requirements and other constraints (The Open Group, 2001) – a similar philosophy underpinning most Agile software development frameworks.

6.10 Full Scale Gantt Chart