

SYSTEMS DEVELOPMENT SCENARIO GENTLEAPPLE

ASSIGNMENT 3

SYD701 SYSTEMS DEVELOPMENT METHODOLOGIES

Dale Stephenson

Submission Date: 06.06.22

Table of Contents

Executive Summary	3
1.0 Methodology	4
1.1 Critical Success Factors for Agile Development	4
1.2 Agile Techniques	6
1.3 Impact of Outsourcing on Agile Methodologies	7
1.4 Deploying Scrum	9
2.0 Elements.....	10
2.1 The Product Backlog	10
2.2 The Increment	12
2.3 The Sprint Goal	14
2.4 The Sprint Review	15
3.0 Rich Picture and Root Definition.....	17
3.1 Rich Picture	17
3.2 Root Definition	18
4.0 High-Level Plan	19
5.0 Application of Elements	23
5.1 The Product Backlog	23
5.2 The Increment	25
5.3 The Sprint Goal	26
5.4 The Sprint Review	29
6.0 Reflection.....	31
6.1 Appropriateness of Methodology	31
6.2 The Use of Elements	32
6.3 Future Application	35
References.....	37

Executive Summary

GentleApple is a mid-sized apple producer, packhouse and cool store enterprise. They have approached the author to develop a cloud-based application and database accessed through a secure web-based user interface (UI). The project's purpose is to update the current spreadsheet-based operational processes and collate them into a single system that is simple to manage and operate, improves data entry input and benefits from an automated backup solution. The system should consider the following requirements outlined in the proposal document:

- Customer account information
 - Customer type
 - Contact details
 - Order details (quality, price, quantity)
 - Order and delivery tracking
- Supplier purchase ordering information
 - Contact details
 - Purchase Orders (item, quantity, price, authorisation)
- Cool store items track and trace
- Employee information
 - Contact details
 - Shift control
 - Annual leave
 - Sick leave
- Integration with the intermediate bin database web API
- Operational performance dashboard

This report outlines the methodology recommended for the development, the justification for the chosen methodology and framework, a high-level indicative plan and practical steps for four framework elements that represent higher value for both business stakeholders and the development team.

1.0 Methodology

The notorious reputation that software systems development has for failure, whether completion on time and within budget or in meeting the objectives of users and stakeholders, means that choosing the most appropriate development methodology is an essential first step to improving the chances of success (Krancher, 2020; Forney, 2019). The criteria of project success are based on organisational needs, to achieve this there has been an increase in leveraging lightweight development methodologies such as Agile over the traditional heavyweight methodologies such as Waterfall (Forney, 2019). Agile is the chosen methodology for the GentleApple systems development by deploying the practices of the Scrum framework, the justification and four main elements form the basis of this report.

1.1 Critical Success Factors for Agile Development

Research conducted on software development indicates the difficulty that decision-makers have when choosing the appropriate methodology (Forney, 2019). The Project Management Institute (PMI) reported in 2018 that mature organisations that achieved a success rate of up to 80% had proven effective at selecting a well-suited methodology aimed at delivering results against goals in terms of cost, time and scope (Forney, 2019). Forney (2019) proposes an *Agile Success Estimation Framework* to assist decision-makers when choosing Agile over other methodologies and address the problems encountered, such as the lack of integrated tools to assess the suitability of the methodology. At this stage of the GentleApple development, the scope is limited to high-level user requirements with a focus on quality as indicated by the documentation and ease of use for a business that employs individuals with comparatively low technical abilities. Cost and time are not stated project considerations, however, it is reasonable to expect these will become important factors during the early stages of the systems analysis process. Furthermore, the likelihood that new information will be introduced during development will increase the need for short decision latency, a factor identified as a leading factor for successful software development projects, increasing by as much as 60% where Agile methodologies are deployed (Forney, 2019).

The methodology decision considers the scenario documentation to define project success by using the *Project Success Criteria* (PSC) classified by the *Applied Project Success Estimation Model* (Forney, 2019):

- Cost/Effort Variance Percentage Success: No Major Overruns - Cost and effortless than or equal to 30%
- Schedule Variance Percentage Success: No Major Schedule Delays – Actual elapsed time in months less than or equal to 30%
- Quality No Extreme Defects Success: Delivered Metric – Zero extreme defects that would cause the software to become unusable within month one from deployment
- Stakeholder Objectives Success: Objectives Met – Described as ‘Fully Met’ or ‘Exceeded Expectation’ in satisfaction survey
- Business Requirements Success: Success Metric – Described as ‘Fully Met’ or ‘Exceeded Expectation’ in satisfaction survey

The importance of the final two success factors is a key consideration for the company acting as the outsourced development partner. The incremental and iterative nature of Agile provides a practice of continuous and rapid feedback loops between the development team and the stakeholders to produce working software over comprehensive documentation (Forney, 2019). Additional success factors have been considered based on assumptions that would be in scope for an outsourced project (Forney, 2019):

- Scope: The ability to produce deliverables, requirements and services at a minimal level
- Project Sponsors: The effectiveness of the project sponsor to manage stakeholders
- Team Talent and Maturity: A diverse team with the skills, competency, self-directed motivation and proven ability to effectively work together
- User Involvement: An ability to include end-users in the process and solicit feedback

Forney (2019) further assesses the positive impact critical success factors (CSF) have on the outcome of software development projects, and the ability for them to mitigate the addition of further complexity or risk factors. In total 28 CSFs have been identified of which 12 were evaluated that covered the following five categories (Forney, 2019):

- 1) Organisation – management, environment;
- 2) People – team capability, customer involvement;

- 3) Process – project definition;
- 4) Technical – delivery, agile techniques;
- 5) Project – type, schedule, nature.

The CSFs identified by the quantitative analysis align closely with the PSC listed in the *Applied Project Success Estimation Model*, providing a measure to support the use of Agile for the GentleApple project. The analysis, however, was more thorough. Of the 12 CSFs analysed only three were identified as significant for project success when deploying the Agile methodology, 1) The strategy for delivery; 2) The software engineering techniques being applied; 3) The capability of the team (Forney, 2019). The three factors overlap with the four important success factor assumptions for outsourced projects, 1) Scope; 2) Project Sponsors; 3) Team Talent and Maturity; 4) User Involvement.

1.2 Agile Techniques

To ensure an effective delivery strategy the Agile techniques must focus on high priority and project critical features first, these include a complete definition of the coding standards before work begins by applying a principle of design through simplicity, delivering rigour for refactoring, producing the appropriate and considered level of documentation with the end-user in mind and applying correct integration testing (Forney, 2019). To consider team capability an assumption is made that GentleApple has deployed the services of the author due to a reputation for meeting delivery strategies. Additionally, the expertise and motivation of the development team benefitting from a proven track record, knowledgeable management with an ability to adapt to change and a high level of technical training are also assumed (Forney, 2019).

Forney (2019) indicates that Agile is capable of demonstrating measured success rates, explaining its popularity. Young (2013) provides a methodology comparison of several techniques against characteristics of project delivery outlined in Table 1. The table identifies seven software development measures, deadline, cost, quality, project size, release frequency, planning and coordination tightness and ranks them from very high (VH) to very low (VL). This table will be used as a countercheck for the methodology chosen for the GentleApple development.

Method	Deadline Emphasis	Cost Emphasis	Quality Emphasis	Project Size	Release Frequency	Planning Emphasis	Coordination Tightness	Notes
LD	H	H	VL	S	H	L	L	
RAD	H	H	VL	S	H	L	L	
DSDM	H	H	M	S	H	M	L	
Internet-Speed Dev	H	H	L	S	VH	M	L	
SCRUM	H	H	M	M	H	M	M	
FDD	M	M	M	M	M	H	H	
Agile Unified Process	M	M	M	M	M	H	H	
Evo	M	M	M	M	M	H	H	
MSF	L	M	H	L	L	H	H	
Open UP	L	L	H	S	L	H	H	
XP	H	L	VH	M	M	H	H	Product quality emphasis
Crystal Methods	L	L	VH	L	L	H	H	Team member utilisation
Spiral	L	L	H	L	M	H	H	Experimental Projects
PRINCE2	L	M	H	L	L	M	M	Focus on process
JAD		M				H	H	Requirements Gathering
Waterfall	L	L	H	L	L	H	H	High cost of rework
PMBOK	L	L	H	L	L	H	H	Standards Compliance
RUP	L	L	VH	VL	L	H	H	Documentation, Training, Certificates
CMMI	L	L	VH	L	L	H	H	Process improvement
SDLC	VL	VL	VH	VL	VL	VH	VH	Formalised process
Enterprise Unified	VL	VL	H	VL	VL	VH	VH	Long-term projects

Table 1: Software development methodology comparison

1.3 Impact of Outsourcing on Agile Methodologies

The challenges of outsourcing Agile development to third parties are identified by specific risks, primarily the interrelationship with the client's business stakeholders (Aoufi et al., 2021). Outsourcing often demands that projects are delivered on a fixed-price contract to an agreed set of requirements and deadlines, which are at odds with the Agile Manifesto's emphasis on an ability to change at the expense of the approved plan (Aoufi et al., 2021). Aoufi et al., (2021) state that digital transformation projects deploying Agile for development are more likely to succeed if guidance is sought from independent consultants or business advisors, allowing organisations to better frame requirements to maximise delivery from the outsourced partner. Qualitative research revealed five challenges that must be addressed when outsourcing Agile development projects:

- 1) Mindset: Organisations perceive themselves to be more capable of implementing, capturing and sharing Agile practices than the reality suggests, very few

organisations can be considered as truly doing Agile (Aoufi et al., 2021). Agile often requires organisational change beyond the scope of the project, deploying the concept of trying but failing fast, a concept that is difficult to grasp outside of the IT sector (Aoufi et al., 2021).

- 2) Understanding: It is important to provide guidance based on evidence for the appropriateness of Agile, especially when clients struggle to separate needs and wants that lead to complications when they change their minds (Aoufi et al., 2021). If there is a lack of clarity and rigour when determining requirements it becomes more difficult to engage in the relationship, increasing the likelihood of failure (Aoufi et al., 2021).
- 3) Preparation: Different sectors display varying maturity with their approaches to Agile (Aoufi et al., 2021). Smaller applications in private enterprises general deploy traditional waterfall approaches for core business processes and Agile for new developments where there is no requirement to consider legacy systems, this heterogeneous approach leads to a mixed operating environment (Aoufi et al., 2021). The approach is reasonable for GentleApple, any legacy systems are spreadsheet data deploying processes that do not provide an adequate quality of data entry thus, the data must be cleansed before being imported to the new system.
- 4) Contracts: Agile contracts are commonly based on a waterfall approach with specific outputs, costs and timeframes with associated penalties for failure (Aoufi et al., 2021). This approach is adversarial and works against collaboration, an important element of software development (Aoufi et al., 2021). Conversely, Agile contracts often avoid commitments on scope, time and process; these contractual differences lead to tension and erode trust between parties (Aoufi et al., 2021).
- 5) Commitment: The development team must focus on supporting GentleApple's management through the phases of the project lifecycle with engagement, understanding and guidance collectively acting as drivers for success (Aoufi et al., 2021). The appropriate resources needed, including access to end-users, should be communicated to maintain the pace of delivery and deploy deliverables into a live production environment, this will require skilful influencers across the organisation and cross-cultural support between parties (Aoufi et al., 2021).

1.4 Deploying Scrum

Scrum has emerged as the most popular of the types of Agile methodology (Forney, 2019). Table 1 outlines the factors that explain this popularity. Forney (2019) identifies the characteristics of Scrum as:

- Short planning cycles of between 7 and 30 days with a focus on requirements and completing tasks
- Ability to produce deliverables effectively within the short iteration (Sprint)
- Prompt incorporation of stakeholder feedback into the following sprint

Scrum is a highly versatile framework derived from its simplicity, consequently, it is deployed in over half of all Agile projects (Hron & Obwegeser, 2018). The literature indicates that the framework and techniques must be tailored to the specific project circumstances and set of requirements to be effective, this may require GentleApple to modify roles and processes as a consequence of an improved user interface (UI) as an output of the development (Hron & Obwegeser, 2018). Scrum inherits the benefits of Agile deployments with the cyclical nature of sprints allowing the team to collaborate and stay well informed through the transfer of knowledge and collective problem solving (BALOG, 2019). The incremental and iterative nature of Scrum also allows a rapid change in direction to adapt to changing market environments, or if a determination is made that the direction is not correct (BALOG, 2019). This adaptability provides the opportunity to manage and reduce risk, particularly at the beginning of the development, and make quick adjustments based on stakeholder feedback and changing requirements (BALOG, 2019).

2.0 Elements

The following Scrum elements have been identified as key to the success of the GentleApple software development, 1) remote outsourced development; 2) technical capabilities of the end-user operating the packhouse; 3) need to test the functionality and usability of the interface incrementally to manage change; 4) receive feedback from stakeholders and end-users early in the design and development phases, mitigating risk and increasing the likelihood of success. The elements of the Scrum framework that encompasses events, artefacts and roles will be implemented to fit the context of the development project whilst staying within the boundaries of the Agile methodology, ensuring the human practice framework remains intact (Shringi, 2020). The elements that make up Scrum are outlined in Table 2 (FUIOR, 2019).

Key Elements of Scrum	
Artefacts	Product Backlog, Sprint Backlog, Increment, Sprint goal
Roles	Product Owner, Scrum Master, Scrum development team, Stakeholder
Events	Sprint, Sprint Planning, Daily stand up, Review and Retrospective meetings

Table 2: The key elements of Scrum

The four most important elements for the GentleApple project, considering the complexity of the development, remote location of stakeholders and users and the resulting identified risks are 1) The Product Backlog; 2) The Increment; 3) The Sprint Goal; 4) The Sprint Review.

2.1 The Product Backlog

The product backlog is an established supporting tool that allows the developers to manage the requirements gathered in the initial phases of the systems development life cycle (SDLC) and prioritise them based on importance as defined by the stakeholders and the technical requirements (Alhazmi & Huang, 2018). Several factors determine the order of backlog items, 1) Priority; 2) Story Size; 3) Complexity (Alhazmi & Huang, 2018). In this sense, the backlog forms the short, mid and long term goals, with short term goals

corresponding to a minimum viable product (MVP) (Shringi, 2020). Responsibility for creating and maintaining the product backlog sits with the product owner, it is important they fully understand the requirements so it expresses the vision for the product (Alhazmi & Huang, 2018; Shringi, 2020). The product backlog provides transparency over the work involved and aids communication of the vision between the development team and the stakeholders, with items categorised as, 1) Product features; 2) Change requests; 3) System defects; 4) Technical improvement; 5) Knowledge acquisitions (Rubin, 2012; Shringi, 2020). The characteristics of a quality product backlog are items that are (Rubin, 2012):

1. Appropriately detailed: Items that should be worked on in the near term will have a greater level of detail, if the work is large or prioritised for later in the development lifecycle these will have less detail. Larger items, such as epics, will break stories down into smaller stories as the team gets closer to the work. It is important not to waste time adding detail to items that may never be worked on.
2. Emergent: The product backlog is never complete or fixed but is continuously developed and emerges over time. This is an important part of agile, allowing the product backlog to adapt to new economic information, unpredictable moves by competitors, unforeseen technical issues or changes made by the stakeholders.
3. Sufficiently estimated: Each item is estimated corresponding to the effort of work involved, large items prioritised at the top of the backlog signal that additional refinement is necessary by the product owner. Estimates should be reasonable but not overly precise.
4. Prioritised: It is not necessary to prioritise all product backlog items, however, near-term items should be adequately prioritised. This would normally include items due to be worked on in the next few sprints, or might extend to all items involved in the next product release, beyond this is not worth the effort. Any new items introduced must be inserted and ordered in the backlog.

There is no expectation that the product backlog is complete at any stage of the development, it is not a fixed set of requirements and tasks, nor is the case that items in the backlog will be completed (Shringi, 2020). The backlog includes items that could be done to ensure a successful end product, such as a feature story or the fix of an identified bug in the code, the team should collaborate in grooming the product backlog with participation from both internal and external stakeholders (Rubin, 2012; Shringi, 2020). The product owner

must make the final decision when grooming the product backlog after engaging in dialogue with the participants (Rubin, 2012).

The product backlog is an essential part of the Scrum framework being used to inform the sprint backlog, which makes it possible to deliver features to the customer sooner to elicit feedback, aid learning, discovery and experimentation, reduce risk, and validate the direction taken (Shringi, 2020). The product owner should be ruthless in their application of the product backlog, having a strong focus on items that produce the most value (Shringi, 2020). To produce a quality product, a defined set of factors should be identified to rank the work involved in the individual backlog items (Alhazmi & Huang, 2018). Planning poker is one technique that, despite having several benefits, relies heavily on expert observation which is not always effective, nor is effort and time estimation, instead task complexity and importance are more appropriate validation criteria for estimation and scheduling (Alhazmi & Huang, 2018). Alhazmi & Huang (2018) recommend the Fibonacci scale as the most effective technique, enabling development teams to focus on tasks deemed most important for a high-quality project outcome.

2.2 The Increment

Scrum is organised into artefacts known as increments and called sprints which, once started, cannot be altered unless there is a compelling reason to do so, such as to obtain more frequent feedback, account for annual holidays at the end of a fiscal year, or to reduce waste (Rubin, 2012). The purpose of the sprint is the completion of a functioning, testable and releasable product (Hernandez et al., 2019). In the context of the GentleApple development, this definition of 'product' is troublesome, in this context the term 'product' is changed to 'a complete and testable deliverable' (Hernandez et al., 2019). Sprints are defined by their short duration, which remains consistent across the project and normally lasts up to a calendar month, timeboxed by a fixed start and end date (Rubin, 2012). The sprint includes several activities, 1) Planning; 2) Execution; 3) Review; 4) Retrospective (Rubin, 2012). A general rule of the sprint is that no changes are permitted that alter the scope or goal of the particular sprint, this allows work to maintain a sustainable pace based on the chosen set of items that align with the sprint goal (Rubin, 2012). Timeboxing the sprint is a time-management technique that has several benefits (Rubin, 2012):

1. Establishes work in progress (WIP): A body of work that is started but not yet completed. A WIP limit on the sprint allows the team to include work they feel capable of completing by the end of the sprint.
2. Forces prioritisation: This allows the team to prioritise the amount of work that matters most, sharpening the focus on work with the greatest value.
3. Demonstrates progress: Completing and validating important work by the end of the sprint allows the team to measurably demonstrate the progress being made, this reduces the risk to the organisation and GentleApple stakeholders.
4. Avoids unnecessary perfectionism: A fixed end date for the sprint reduces the human desire to aim for perfection when 'good enough' is a satisfactory output.
5. Motivates closure: A fixed close date motivates teams to close the work, encouraging them to diligently apply themselves to the assigned tasks through the deployment of a hard deadline. No end date reduces the sense of urgency.
6. Improves predictability: It is far easier for the team to predict with a high degree of certainty the work necessary in the short sprint cycle, this is far more difficult over longer periods.

The short duration sprint cycle means that planning the work is far easier and requires significantly less effort (Rubin, 2012). This benefit extends to the quick turnaround time of feedback from stakeholders by providing an opportunity to inspect and adapt the software being developed and remove undesirable product features before a bad decision is compounded (Rubin, 2012). The resulting speed of the feedback gained improves economic viability by providing an opportunity to generate revenue sooner and improving the return on investment (Rubin, 2012). GentleApple currently experiences frequent errors from their existing spreadsheet data entry process, the sooner these data processes are incorporated into a constrained system that accounts for redundancy, the greater the quality of the service and reduction in employee administration of the data. Shorter sprints will also mitigate the effect of errors by bounding them to the sprint timeframe, this is also true if a path has been followed that is wrong, the team only need to unwind the work carried out in the previous sprint (Rubin, 2012). In effect, the sprint acts as a meaningful checkpoint or milestone to demonstrate the working features of the development on which to base decisions and ease the management of the complex development environment through more actionable opportunities to inspect and adapt (Rubin, 2012). Shorter sprint cycles have an added benefit

for team morale by rejuvenating excitement in the development through gratification of the work completed, this boosts enthusiasm for the project and increases the likelihood of success (Rubin, 2012).

2.3 The Sprint Goal

The length of the sprint iteration enables the sprint goal to be discussed, the sprint goal typically describes the business purpose and the value expected from the increment (Rubin, 2012). To achieve the greatest value, the sprint goal should be clear and contain a single focus acting as a defining output of the sprint planning, this should be discussed to achieve mutual commitment as it will be used to determine the product backlog items to be moved to the sprint backlog (Rubin, 2012). Mutual commitment and the ability for the sprint backlog items to be achieved by the end of the sprint, whilst balancing the ability to adapt to change, is essential (Rubin, 2012). The mutual commitment means the goal must not be *changed* during the sprint, that is not to say further *clarity* cannot be sought, this difference should be understood to avoid undesirable consequences (Rubin, 2012):

- Change: Any alteration that can generate economic waste that is meaningful to the workflow or scope. Practically speaking, this might include removing backlog items from the sprint or fundamentally altering the scope of a backlog item. If a change represents significantly more effort by the team, such that it might impact the ability to complete the items in the sprint, then it should be avoided.
- Clarification: This might be additional details provided to assist the team in achieving the sprint goal, such as adding specificity to backlog items previously unknown at the start of the sprint. It is reasonable to seek clarity during the sprint and the product owner should answer any questions asked.

The inability to change a sprint goal does not break the important embracing of change, instead, it recognises the economic effort invested in the readiness of the backlog items (Rubin, 2012). Any changes at this stage negate the work committed and will require further sprint planning to alter the goal, furthermore, once the sprint execution has begun and backlog items have transitioned through the Kanban board, the investment of work increases (Rubin, 2012). For instance, if the team have started work on feature 'A' but this is changed

to feature 'B', which is not specific to the sprint goal, even if feature 'A' is continued in a later sprint, there is still planning waste, if there are dependencies to other backlog items the waste is amplified (Rubin, 2012). Waste only increases from this point if feature 'A' happens to have been completed or time is required to remove partially completed work (Rubin, 2012). The effect of waste has wider implications on the team, eroding trust in the product owner for violating the commitment and leading to a deterioration in motivation that can have a longer-lasting impact on team diligence with future backlog items (Rubin, 2012). The relationship between the development team and GentleApple must be carefully nurtured to mitigate the risk of changes occurring to the sprint goal. Those stakeholders with influence must understand the implication that changes will have on the team and the long term success of the project. Internal business considerations should also be accounted for, such as reputation damage and legal ramifications that, at the extreme, might lead to a breach of contract terms. External factors may also impact the sprint goal such as the actions of a competitor that necessitates material change (Rubin, 2012). However, in the case of GentleApple and within the scope of the development, this is unlikely. If despite the best efforts of the team the sprint goal is invalidated, a decision can be made to abnormally terminate the sprint, leading to a sprint retrospective between the product owner and development team to decide on a new sprint goal (Rubin, 2012).

2.4 The Sprint Review

The sprint review should take place towards the end of each sprint to inspect the results of the work achieved during the sprint execution, with an analysis of the current state of the product (Rubin, 2012). Sprint reviews prevent the project development from going off track leading to disappointment and frustration, which can happen when reviews take place over longer periods (Rubin, 2012). In addition to the internal Scrum team, external stakeholders from GentleApple should be encouraged to attend the sprint reviews and offer valuable feedback. If it is not reasonable or necessary to physically attend at these regular intervals, virtual attendance options should be provided and meetings recorded for later review. The sprint goal can be used as an indication of which stakeholders are likely to benefit from attending, in any event, completed work should be presented meeting the established definition of done (Rubin, 2012).

A definition of done must be agreed with GentleApple to enable the development team to determine if the build is potentially shippable (Rubin, 2012). In the context of this development, it will not be possible to deliver a shippable product after each sprint, there will be a significant amount of materially important backend work to be completed, particularly in the early phase of development. Scrum permits the definition of done to evolve as the development progresses (Rubin, 2012). Once it becomes practically possible to yield a slice of product functionality that is valuable to the customer, the bare-minimum definition should include design, build, integration, testing and documentation (Rubin, 2012). To determine if the definition of done has been satisfied each backlog item must meet the checklist, including any specific acceptance criteria that necessitate acceptance testing by the product owner (Rubin, 2012). The product owner accepts only those backlog items that meet the definition of done following a just-in-time review, this allows them to answer relevant questions during the review meeting and prepare the demonstration (Rubin, 2012). Demonstrating the work should not be to create a buzz, but merely provide sufficient transparency for inspection and adaptation in an informal setting, it should not demand significant time or effort to present (Rubin, 2012). The scope of the review meeting should be restricted to those artefacts produced during the course of the sprint execution as chosen to meet the sprint goal (Rubin, 2012). If discrepancies exist between what is presented and the sprint goal, the team will be expected to provide explanations that justify the results, the environment must be free from blame allowing the information to be used to determine the next best course of action (Rubin, 2012).

An important output of the spring review is to obtain a clear picture from the stakeholders by asking questions such as, are they happy with the direction, do they want changes, is it meeting specific requirements, have we overlooked a desired feature, is too much work being invested in a low priority feature (Rubin, 2012)? The ability of sprint reviews to adapt the direction of the development should not be underestimated, for instance, a high-level stakeholder might disagree with the authority of the product owner and the application of the definition of done, where this occurs it should be discussed (Rubin, 2012). Furthermore, the criticality of the sprint review should be communicated to avoid sporadic attendance that could lead to dysfunction and commitment in the wrong area (Rubin, 2012).

3.0 Rich Picture and Root Definition

3.1 Rich Picture

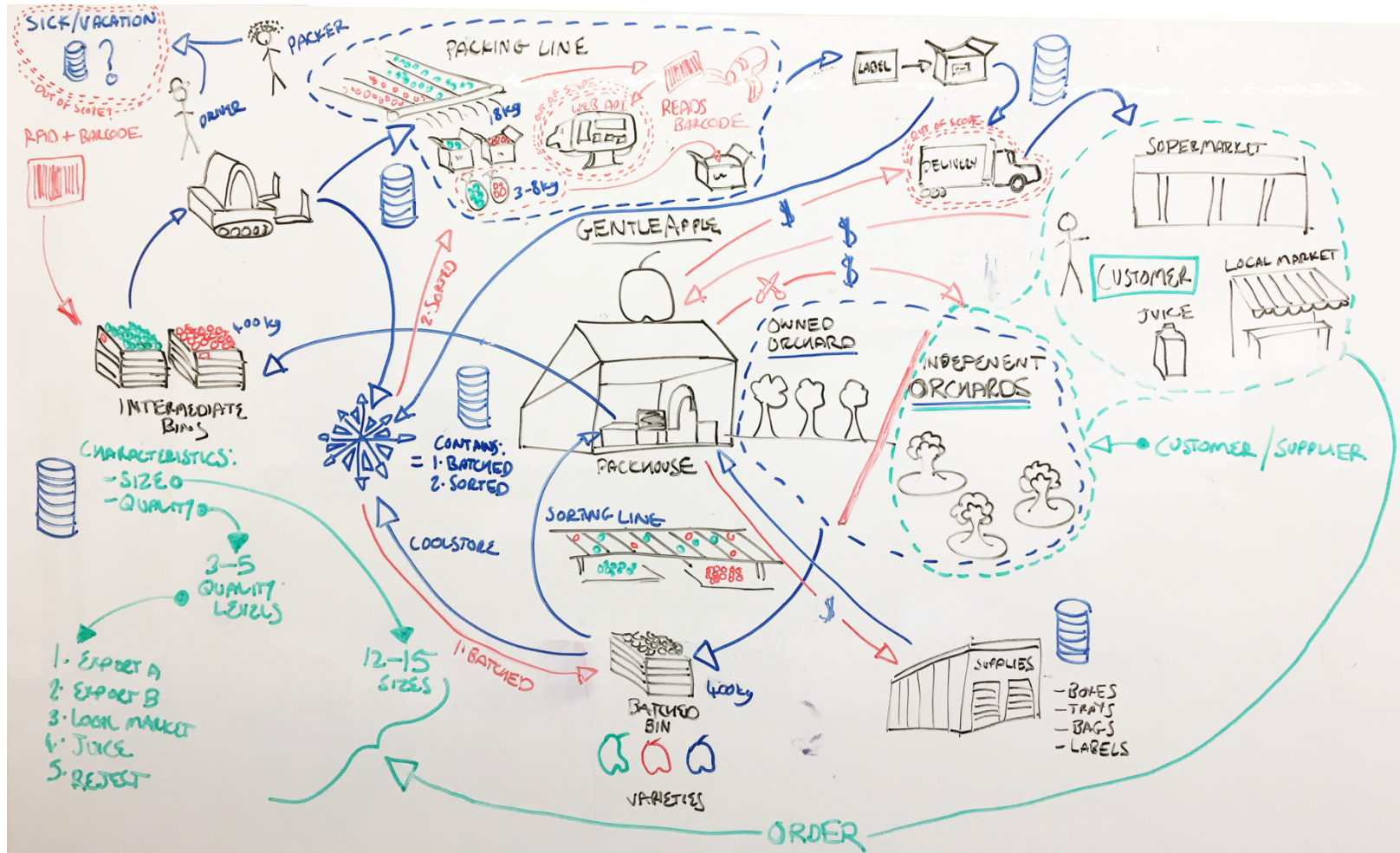


Figure 1: Rich Picture for GentleApple scenario

Several issues have been identified by the creation of the GentleApple Rich Picture:

1. Clarify the customer base from the end product customer and supplier or raw materials customer, what are the differences in data entry?
2. Can the logging of sick and vacation leave make sense in the context of the application development, or should this be scoped out with another solution or the existing solution?
3. There are several inputs into the cool store at each stage of the packhouse process, there is a need to understand how the inventory stages are tracked
4. Can the RFID and barcodes be used to effectively track the product as it progresses and can this reduce or eliminate data entry errors?
5. Identify those employees that carry out the data entry, interviews should take place to understand the level of technical ability of these users to inform the UI design, specifically the data entry process and any dashboard reports.
6. Do the apple quality characteristics change or are they fixed, are they defined differently for each customer?
7. Can digitalisation of the process improve the business operation, do opportunities exist to make data entry less onerous through integrations with existing systems?

3.2 Root Definition

GentleApple provides a fully managed packhouse facility to sort, grade, store and pack apples from both independent and company-owned orchards, the packing line deploys both autonomous processes in conjunction with manual activities that relate to quality control, product grading and the transfer of goods in various states to the cool finishing with separate packing options for a variety of end-user customers.

4.0 High-Level Plan

The development phases outlined for the GentleApple software development are not intended to be fixed, this would require onerous guesswork at the expense of learning and continual development as the system progresses (Rubin, 2012). The high-level phases that will be deployed for the GentleApple development are based on the Scrum structure with consideration made to the mid-sized status of the packhouse operations. Specific practices and techniques will be utilised in the following phases, 1) Initiate; 2) Plan and Estimate; 3) Implement; 4) Review and Retrospect; 5) Release (Sliger, 2011). An assessment of the readiness of GentleApple to accept Agile as the methodology will be made, including their ability to provide resources and access to key employees who are capable of supporting the product development. Table 3 outlines a high-level plan integrating the Scrum practices as they relate to each phase, it includes details regarding the roles, activities, artefacts and associated rules, the list specifically excludes the *Scrum of Scrums* due to the project scope and size not requiring multiple Scrum teams (Rubin, 2012).

Phase	Practices, Techniques and Tools	Outputs
Initiate – Creating the vision of the product	Work with the management of GentleApple to document a vision for the application including the problems it should solve.	A clearly defined vision is provided by the business with a set of features ordered by importance (Sliger, 2011).
Initiate – Identifying roles and stakeholders	Assess candidates for ScrumMaster and Product Owner from the outsourced team. Discuss the meetings that have taken place with GentleApple management to recognise likely supporters of the project within the organisation.	Identify the ScrumMaster, Product Owner and key stakeholders in GentleApple that will support the process and advocate for the project, likely to be those users confident in using technology and department heads directly benefiting from the system.
Initiate – Formation of the scrum team	Select members of the software development team who have the necessary skillset and expressed motivation to work on this type of project. In addition to key roles, the team will require cross-functional expertise, valuing independence with few dependencies (<i>High-Performance Teams</i> , n.d.).	The people who are jointly responsible for delivering the application (Sliger, 2011). Specialised skills are necessary, all team members will be referred to as developers (<i>Scrum Team Roles and Responsibilities</i> , n.d.). Selection includes a database expert, one back-end developer, one front-end developer and a UI designer. The team including ScrumMaster and Product Owner total 6.

Initiate – Develop the Epics	Assess the development project to identify epics that provide a high-level representation of large activities that have measurable economic value (Rubin, 2012). Epics added to Scrum board and accessible to the entire team.	Epics that are driven by the business requirements: 1) Customers; 2) Orders; 3) Inventory/Supplies; 4) Cool Store Tracking; 5) Relevant Sorting Line Data; 6) Employee information (Investigate purchased solution); 7) Access anywhere, cloud infrastructure.
Initiate – Create a prioritised backlog	Identify backlog items that have greater priority or are necessary for early completion in the product life cycle. Near-term items destined for the next several sprints are prioritised (Rubin, 2012).	Work should focus on creating an MVP as soon as possible in the product life cycle, the architecture must be designed and developed with enough back-end and front-end capability to test a high-priority feature based on business needs whilst considering relative ease to complete.
Initiate – Conduct release planning	Plan for incremental releases for faster feedback and to improve return on investment (ROI) (Rubin, 2012). A release plan that allows for learning and pivoting when necessary (Rubin, 2012).	For a complete breakdown of the project, it makes sense to start with the data separated by the spreadsheets. To increase the rate of progress and ROI the customer spreadsheet could, for example, be further dissected into the customer types.
Plan and Estimate – Create the user stories	Craft user stories that are understandable to business and technical users, and consider their granularity and ease of refinement (Rubin, 2012).	A set of structurally simple user stories that express business value for many backlog items (Rubin, 2012). Starting with the customer spreadsheet would make sense, the data will be simple to clearly define and migrate.
Plan and Estimate – Approve, Estimate and Commit user stories	Assess the value of the user stories, the conditions of satisfaction expressed through acceptance tests, and provide an estimation of work (Rubin, 2012). A commitment to user stories that add value to the customer, end-user, or both (Rubin, 2012).	Once incremental releases have been identified user stories that represent the high-level view should be discussed with the stakeholders and approved. It should be made clear that these are not fixed and can be adapted to meet changing requirements as the project develops.
Plan and Estimate – Create tasks	Craft user tasks that make up an epic representing a collection of related stories that provide a shared understanding of the build order (Rubin, 2012).	A sequence or common workflow of user tasks (Rubin, 2012) defines the actual work that must be completed to meet the user stories and epics.
Plan and Estimate – Estimate tasks	Assess tasks based on the effort required to complete them using story mapping (Rubin, 2012).	Tasks logged with story points defining effort based on the Fibonacci method. These should be considered based on the understanding of the GentleApple project and previous experience.
Plan and Estimate – Create a	Summarise the sprint goal by describing the business purpose and value of the sprint (Rubin, 2012).	An agreed sprint goal containing backlog items to be completed by the end of the sprint (Rubin, 2012), the prioritised

sprint backlog		product backlog can be used to create the sprint backlogs.
Implement – Create the deliverables	Define the incremental deliverables capable of meeting the short duration sprints (Rubin, 2012). A set of deliverables subjected to the environment they must exist (Rubin, 2012).	All members of the development team must, in conjunction with the business stakeholders, clearly understand what the expected deliverables are and the work involved. If the team is not confident they can be met further discussion should take place.
Implement – Conduct daily stand-up	Establish daily sprints, preferably at the same time, to promote brevity through the practice of standing up (Rubin, 2012).	A daily stand up for all development team members timeboxed to 15 minutes or less (Rubin, 2012). Stakeholders should be invited and encouraged to attend these meetings virtually.
Implement – Groom the prioritised product backlog	The product owner leads a collaborative effort to create, refine, add details, estimate and prioritise backlog items (Rubin, 2012).	A groomed product backlog that ensures items at the top is ready to be moved into a sprint (Rubin, 2012). The development team should support the grooming process.
Review and Retrospect – Demonstrate and validate sprint	Demonstrate the work the team has achieved during the sprint and the burn-down chart, providing transparency and excitement (Rubin, 2012).	An examination of the product confirms the sprint work is complete and recommends changes (Rubin, 2012). It is essential business stakeholders attend this meeting and provide feedback and direction for the team.
Review and Retrospect – Retrospect Sprint	The development team, ScrumMaster and Product Owner inspect the process and consider ways to adapt (Rubin, 2012).	An understanding of what is and is not working, focusing on continuous improvement (Rubin, 2012). Business stakeholders do not need to attend this meeting, however, all the development team should be present.
Release – Ship deliverables	Deploy the product backlog to deliver business value software development in the presence of uncertainty (Rubin, 2012).	Rapid and regular product and feature updates for testing according to the definition of done (Rubin, 2012).
Release – Retrospect project	The development team, ScrumMaster, Product Owner and stakeholders meet to reflect and propose improvements (Rubin, 2012).	Create a set of improvement actions based on the available data and close the retrospective (Rubin, 2012).

Table 3: Phases, Practices, Techniques, Tools and Outputs of the GentleApple development

An analysis of the project factors outlined in Table 1 has been conducted to inform the decisions made for the high-level plan:

- **Deadline:** The scenario does not identify a timeframe for the project, however, given the errors encountered by GentleApple's current use of spreadsheets and the desire to deliver quality service and products to customers, it is reasonable to expect a deadline will be negotiated. Scrum is a methodology with a high deadline emphasis designed to meet frequent release dates (Young, 2013).
- **Cost:** It is reasonable to assume that the development will not be a 'money is no object' scenario. It is common for businesses to experience cash flow restrictions and a desire to build improvements into processes in a cost-effective way. Scrum has a high emphasis on cost consideration supporting the outsourced company to manage the budget at the expense of other features (Young, 2013).
- **Quality:** GentleApple places great emphasis on quality and Scrum achieves medium-level quality development. In this case, quality refers to a robust, bug-free piece of software that, as shown in Table 1, comes at the expense of deadline and cost (Young, 2013). A medium emphasis here is considered reasonable given the expected importance placed on deadline and cost.
- **Project Size:** Although an important piece of software for GentleApple, the project would be considered small, requiring fewer than ten developers, the medium emphasis ranking is sufficient (Young, 2013).
- **Release Frequency:** Release frequency is important to allow for user testing and feedback, however, the system is not public-facing or experiencing iterations in days, rather, it will be weeks or months between versions (Young, 2013). The high emphasis is more than acceptable for this project.
- **Planning emphasis:** It would be unreasonable to expect a lengthy planning phase for a project of this size. However, an overall design plan will be required as the project size will not necessarily be a construction of the separate components (Young, 2013). Medium emphasis is sufficient.
- **Coordination tightness:** An overall design plan will require a reasonable degree of tightness between a relatively small team, the components will not be built independently from one another but will be organised in order by how they interconnect (Young, 2013). Medium emphasis is adequate for the project purpose.

5.0 Application of Elements

5.1 The Product Backlog

Establishing the product backlog for the GentleApple project should be considered foundational, providing a structure from which to start development based on a shared understanding of the tasks involved that add tangible value to the stakeholders and begin user testing early in the process. Outsourcing of product development means that access to end-users and stakeholders will be more challenging. Deploying a Kanban board in a tool such as Jira Service Management aids visibility and communicates the steps involved to produce quality software that is capable of meeting the needs and technical competency of the GentleApple management and users (Zayat & Senvar, 2020). Kanban and Scrum have similarities that mean they work well together, breaking up complex tasks into smaller user stories, aiding self-organisation and management, both can work to customer requirements and place a high value on optimisation and continual improvement (Zayat & Senvar, 2020). The product backlog should be applied as a validation tool in the daily scrum meetings to measure the goal to be achieved (Kurnia et al., 2018). To support the establishment of the backlog the initial planning phase should be used to discuss the overall architecture of the system, a decision can also be made about the platform and technology that will be deployed and combined these will inform the product backlog (Moe et al., 2010). During this phase no code will have been written by the GentleApple development team, they may find this frustrating as they see it as being a period of little progress (Moe et al., 2010).

The product owner should carefully manage the GentleApple product backlog, acknowledging the collection of work items as evolving throughout the project and requiring constant refinement and grooming (Sedano et al., 2019). The product backlog will be used to sum up the total amount of work remaining to reach the GentleApple development goal (Sedano et al., 2019). Moe et al. (2010) reported that any division of work can lead to developers creating their own plan with little team discussion, this type of decision hijacking highlights a developer's ability to prioritise personal objectives over team goals that can lead to a lack of trust in the development process. All team members must have a clearly defined picture of the GentleApple development, the product backlog must act as a boundary object for the GentleApple development team (Sedano et al., 2019). For instance, it does not make sense for the team to develop functionality for employee sick leave and vacation time, this

should be out of the scope given the many applications available to purchase that can perform this separate and common business process. The GentleApple project backlog should include user stories, identified in Table 4, the order is indicative but can act as a guide:

User Story	Story Point	Priority
As a user, I need to input data in a way that does not cause duplicates and reduces errors	3	1
As a Manager, I need reliable data that can be cross-referenced easily with other data sets, such as the level of supplies compared with new customers, to gain a better overview of the business	5	3
As a user, I need to update information once so it is applied across the data sets reducing the time spent on data input	1	1
As a forklift driver, I need to know when cool stores are reaching capacity so I can utilise the capacity available more effectively	8	2
As a Director, I need confidence that data is being backed up to ensure we maintain our operational history	3	1
As an admin, I need to be able to offboard employees when they leave the company to ensure the security of the data	5	1
As a user, I need automatic spell checking to minimise data entry errors	2	4
As a Packhouse Supervisor, I need to be notified when the sorting, packing line and cool store reach capacity so I can allocate resources	13	1
As the Financial Controller, I need to produce business reports for auditing and management accounting	5	3
As a remote worker, I need to be able to access the system anywhere and on any device	3	1

Table 4: Indicative Scrum Product Backlog deploying User Stories

5.2 The Increment

The GentleApple business model is a relative uncertainty to the development team, to mitigate risk and improve feedback time from stakeholders a shorter sprint cycle of two weeks is recommended. A one calendar month sprint cycle is likely to negatively impact communication of the progress made, conversely, a one week cycle would limit the amount of work that can be achieved by a team of this size and the potential of them failing to produce a shippable product. The regular increment duration will provide the team with a cadence to the development effort, this rhythmic familiarity will allow the team to get 'into a groove' of habitual activities such as the daily stand-up and sprint backlog, making it easier to stay mentally on track (Rubin, 2012). A two-week increment length is capable of directly impacting the team's collaboration, meetings will not be so frequent they risk becoming unproductive, but frequent enough to provide a sufficient level of feedback and observation from stakeholders and end-users, positively impacting progress and increasing the likelihood of success (Moe et al., 2010). All of the work achieved by the development team as part of each increment will be represented by the true state of the product at the end of the sprint, the sprint review should be used to inspect the work achieved for the GentleApple project (Fowler, 2019). The output of the increment will allow the development team and GentleApple stakeholders to make better-informed decisions about what is being developed to generate new ideas and recommend changes (Fowler, 2019). The increment must produce work that is done and can be delivered to GentleApple with no further work required, requiring the team to be highly organised as research shows this is a difficult threshold to achieve and maintain in practice (Fowler, 2019). A determination should be made that describes when the product of work is materially shippable, Table 5 is a checklist that should be used to define the definition of done (Rubin, 2012).

The definition outlined in Table 5 accounts for the nature of the product being built and the technologies used, the sprints can be measured and used to show progress to stakeholders, regardless of whether the product is shippable. The intention of the definition is not to excuse the inability to deliver validated value to GentleApple; it merely acknowledges the ground up modernisation of the data entry process moving away from spreadsheets. This should be supported by the prioritisation of items in the product backlog to emphasize product delivery for end-user testing early in the development lifecycle. Two-week increments would be sufficient time to generate incremental work that falls within the definition of done, requiring

no further work. The team must remember that in Scrum it is not considered possible to measure the amount of work that has not occurred (Fowler, 2019). Although two weeks is recommended, the increment should be negotiated first between the product owner and development team for two reasons, 1) the product owner often values shorter sprint increments and the transparency that comes with the end of a sprint, whilst 2) developers value longer sprint increments with a focus on the stability between the beginning and the end of the increment (Fowler, 2019). The recommended two weeks may ultimately not be decided by the team as feasible, however, providing it remains within the Scrum rules of no longer than 30 calendar days, it will be approved (Fowler, 2019). It must be noted that the sprint increment is not a deadline, the increment should never be extended to provide extra time to complete tasks, the increment is a measuring tool to help the team to learn and improve as the development progresses (Fowler, 2019).

Definition of Done
Design reviewed with GentleApple end-users
Code completed <ul style="list-style-type: none"> - Code refactored - Code in a standard format - Code is commented - Code checked in
End-user documentation updated, training or user manuals, troubleshooting (Where required)
Tested <ul style="list-style-type: none"> - Unit tested - Integration tested - Regression tested - Platform tested
Zero known defects
Acceptance tested (Technical / End-user)
Live on production servers (Where possible)

Table 5: Indicative definition of done to close sprints

5.3 The Sprint Goal

The sprint goal should be agreed upon by the team and business stakeholders during the sprint planning meeting with the help of the ScrumMaster, GentleApple must be invited to the discussion (Fowler, 2019). The goal should be clear and to the point, meaning a short,

high-level brief (Fowler, 2019). The sprint goals may be defined in the following indicative ways:

- We would like to have the database built and deployed in the development environment to determine if spreadsheet data can be ingested
- We would like to create an MVP version of the end customer data entry user interface to get beta end-user feedback
- We would like to connect the development database to the existing sorting and packing line software

Focus and clarity for all developers and stakeholders is an important outcome of a clearly defined sprint goal, particularly where an MVP is considered as the development output (Fowler, 2019). The sprint goal is particularly important for an outsourced project to help business stakeholders anticipate what to expect from the sprint.

The sprint goal provides another important function that may be necessary given the risks of the GentleApple development, providing the only means for cancelling the sprint before it expires, this would occur if the team determines that there is no longer any value in proceeding with the goal as it is outlined (Fowler, 2019). The development team should take the practical steps in the event a sprint goal is invalidated, there are three options (Rubin, 2012):

- 1) Make the new sprint goal fit with the time left of the terminated sprint to keep the team synchronised with the original cadence;
- 2) Make the new sprint the normal sprint length ignoring the time assigned to the terminated sprint, this keeps the scope of the sprint goal uniform;
- 3) Increase the length of the next sprint to account for the time left in the terminated sprint, increasing the scope of the sprint goal.

Considering the context of the GentleApple development, option one or three would be acceptable, with a decision based on the amount of time left in the terminated sprint. The team should then move into the next full sprint with a defined sprint goal that would be reasonable within the scope of a normal goal and time frames should be set to those expected, this will help external stakeholders stay within the original sprint increments. It should be

noted that the short increment of the sprints will reduce the risk that a sprint could be invalidated and the experienced team should be capable of deploying less drastic measures (Rubin, 2012).

The sprint goal should be used by the product owner and developers to inspect the backlog and select tasks that help them achieve the goal (Fowler, 2019). The product owner is best suited to communicate the problem that must be solved and its degree of importance to GentleApple stakeholders, conversely, the development team can provide their input and outline any technical difficulties they expect to encounter including any dependencies that exist that might make reaching the goal a challenge (Fowler, 2019). Ultimately, the team should agree on the set of items that are most likely going to achieve the greatest value to GentleApple, focusing on practical development within the constraints of the sprint (Fowler, 2019). Furthermore, the product owner should fully understand the capacity of the team to ensure the sprint scope has a chance of success, the product owner should determine (Fowler, 2019):

- The size of the work each product backlog item contains;
- The team capacity level that each item will require;
- The overall teams capacity to achieve the work;
- At what point the work becomes too great.

The product owner should base their conclusions on empirical measurements and not predictions, this will become easier as the GentleApple project progresses (Fowler, 2019). In the first instance, the product owner should use measurements from previous projects the team have collaborated on to establish base metrics. Once the team's velocity on the GentleApple development is known, it can be compared to provide better-informed decisions (Fowler, 2019). Story points using the Fibonacci scale can support this process by combining the results of the team based on experience levels and calibration of the actual work produced, the points scale can be assigned to backlog items to increase the likelihood that the work selected will achieve the sprint goals (Fowler, 2019).

5.4 The Sprint Review

The sprint review should include the entire development team and GentleApple stakeholders to provide valuable feedback, ideally, this should be done in person for communication ease with provisions made for virtual attendance where this is not possible. The demonstration of sprint work may not always be easy, particularly in the early stages of the GentleApple development. It is a commonly mistaken view that the review should be akin to a final quality control review, the team should not fall into this trap (Fowler, 2019). For example, if work achieved relates to a sprint goal and corresponding product backlog items that focus on code applied to back-end architectural development, it might make little sense and be impractical to present (Rubin, 2012). However, in such cases the product owner should, at a minimum, include backlog items that relate to testing, this testing can be demonstrated and is a valid output that can also be presented to non-technical stakeholders (Rubin, 2012). Failing this, other communication tools might be considered appropriate to show the progress towards completed work and increase confidence in the development, such as the sprint burndown chart, which is simple to interpret and easy for non-technical users from GentleApple to understand (Rubin, 2012). Sprint reviews should allow the team to identify coding errors and integration problems, at first this may be chaotic and be less than satisfying, however, once the business stakeholders begin approving and communicating their satisfaction with the development team the process will improve (Moe et al., 2010). Moe et al. (2010) highlight the importance that team members feel their contribution to the sprint review is valued to prevent them from withholding information in the future, the product owner and ScrumMaster must manage their response to problems in a way that avoids the team withdrawing if they believe the response is an overreaction.

After the work has been demonstrated by the development team the product owner should accept or reject the work output based on how well it conforms to the requirements and sprint goal (Fowler, 2019). If the work is accepted it can be considered for release to GentleApple at a time of the product owners choosing (Fowler, 2019). If work is ultimately rejected, the product owner must put the items back into the product backlog and reprioritise for a future sprint (Fowler, 2019). It is important that consideration is made of the item's importance to the business stakeholders to ensure requirements continue to be met and stakeholders receive regular communication of work that is done (Fowler, 2019). Should items be completed before the end of the sprint meeting the definition of done, the product

owner can make a business decision to release the work to GentleApple immediately (Fowler, 2019). Fowler (2019) states that the sprint review is a discussion of the increment irrespective of whether or not a portion of the product is already in production and the hands of GentleApple. After the sprint review, it should be clear to all attendees what the product is now capable of doing and what else it could do (Fowler, 2019). The product owner should use the sprint review and inspection of the work product achieved during the increment to generate ideas to make the product more valuable, these ideas may generate new product backlog items, the product owner should ask GentleApple stakeholders for their input with this process by engaging in an open discussion (Fowler, 2019).

6.0 Reflection

6.1 Appropriateness of Methodology

The evidence and literature referenced in this report identify Scrum as the appropriate framework within the Agile methodology for the GentleApple development. That is not to say other frameworks would not be equally suited, Table 1 highlights several other frameworks that would have a sufficient mix of factors identified as medium or higher that might likely lead to project success. It is important to note that although Scrum has been identified as the primary framework deployed for development, the plan for GentleApple includes several other techniques and tools associated with the broader frameworks that represent iterative and incremental development, for instance, the application of user stories for the product backlog and the concept of simple design, which would be beneficial given the technical competency of GentleApple employees. Furthermore, the development plan has discussed the iteration velocity associated with user stories during the iteration, a method more generally associated with Extreme Programming. Despite deploying techniques that do not formally exist within the Scrum framework, the literature suggests that Scrum is flexible and adaptable enough to accept techniques if they support the development of the system and aid the success of the product, providing they stay within the rules of Scrum. The integration of techniques from other frameworks is not to diminish the effectiveness of Scrum, but merely to acknowledge the low maturity level within organisations practising Agile methodologies (Pereira & de FSM Russo, 2018). Pereira & de FSM Russo (2018) suggest that a *Design Thinking* (DT) approach based on *Human Centred Design* (HUD) within Agile can build solutions that are desirable for end-users and support the identification of problems, whilst being technologically feasible and viable for business deployment. This process is achieved through visual representations of the development, making it tangible and easier to accept. The process can also fit neatly into the iterative and adaptable nature of Scrum should new components be identified (Pereira & de FSM Russo, 2018).

The GentleApple scenario may not lend itself entirely to Scrum with regards to the production of documentation. Agile methods prefer the development code to act as the documentation in conjunction with the code comments, given the uncertainty provided by the scenario documentation regarding the ownership of the end product, developers must provide sufficient explanations that are appropriate for the level of complexity within a given piece of code (Cho, 2008). Cho (2018) provides evidence indicating that developers find it difficult to

work on parts of the system that they are unfamiliar with and where documentation is required, this justifies the GentleApple plan to deploy a small Scrum development team to mitigate this likely problem (Cho, 2008). This may still be problematic if there is a change in developers during the project, or new developers come to it months or even years after deployment, this leads to developers becoming overwhelmed and having to ask a lot of questions from those developers with more experience with the application, which can result in a loss of development time (Cho, 2008). The problem is more pronounced where the application might be transferred to another outsourcing company in the event the relationship breaks down or expertise with the project is focused on one individual, leaving one significant point of failure should the level of documentation not be sufficient (Cho, 2008). As an outsourced project Scrum does not always provide techniques and tools to gain the desired level of feedback from the business stakeholders, the reason is often that stakeholders have their regular daily job that can limit involvement (Cho, 2008). Cho (2018) suggests that customer involvement can end up being as little as twice in each sprint, meaning decisions are not made until after the increment work is complete. This is likely to be a significant problem for the GentleApple development for two reasons, firstly, development is taking place away from the business users and business operation and secondly, GentleApple may believe they are being clear with their defining of requirements, but a lack of clarity held by the developers surrounding the business model may still result in misunderstandings. Although Agile and Scrum are adaptable to new information, a lack of customer involvement is not ideal to maintain the pace of work or the quality of the outcome.

With the advent of Covid-19 the work environment should be considered, Cho (2018) indicates that remote working can make it difficult to find a suitable time for all developers to attend the regular Scrum meetings. When a meeting does take place, it can often become inefficient due to team members 'catching up', this can be further exacerbated when the project is in a low complexity phase. Despite these shortcomings, the industry has identified Scrum as the breakout framework for deploying Agile methodology for software development and is appropriate for the GentleApple scenario.

6.2 The Use of Elements

Each element of the Scrum framework has its place, however, the elements chosen in this report highlight the unique aspects of the GentleApple scenario. In particular, the risk

factor associated with development being outsourced and the benefits of Scrum for a project that is not produced locally to the business stakeholders. The decision for each element was based on the ability to increase involvement by GentleApple stakeholders and users and improve communication between the separate companies.

1) *The Product Backlog* aims to prioritise the needs of the business through the deployment of a set of identified tasks and user stories that can be drafted into a readily available cloud-based SaaS tool. As a software development company this is likely already in use and highly familiar to the development team. The product owner can easily draft the user stories and high-level tasks acting as an effective and transparent tool providing restricted access to the business stakeholders. This can increase confidence in the development team's ability to understand the problem domain and fill in any gaps as necessary. This does not only satisfy the Scrum practice of transparency but also aids communication, a challenge the literature has identified with these types of remote to the end-user projects. The familiarity of the development team with this process and the tools will make them easy to use, the relative simplicity means it would not be difficult to provide user training for the business stakeholders, however, lower-level system users are unlikely to find the system as easy to adapt to. Furthermore, given the job roles, access to a computer and the decision making powers these users have means it is unlikely they would require access to this level of development detail. There are some drawbacks to the product backlog being used in this way, business stakeholders need to understand that the work it contains is not guaranteed for project completion and merely indicates the work that might be achieved. For the business stakeholders, the focus should be on the product backlog items prioritised towards the top of the backlog that they can help to assess, ensuring they benefit from the greater level of detail and applying story points to help them understand where the difficulty lies. Although there is a not inconsequently degree of work involved in the product backlog, it will support the entire team throughout each sprint cycle.

2) *The Increment* seeks to mitigate the risk by delivering work to stakeholders sooner, which may be difficult to realise. The literature suggests stakeholder involvement in the work process during the sprint is challenging, there is little evidence from the GentleApple scenario to suggest that this project will be any different. Despite this, setting the right increment time frame is crucial to maximise use, too long and the GentleApple team may become discouraged or lose interest, too short and they may feel overloaded with the process and feel fatigued with the decision making process. The increment is an easy element to establish once the period has been decided and it can be expected that the development team

will use it well. Once the GentleApple business stakeholders become used to the rhythm it is hoped they would find the increment length equally useful, knowing when work is being delivered and arranging their timetables around events, such as the daily stand-ups or sprint reviews that are important to them. It likely will not entirely suit the GentleApple way of working in the same way it does the development team, however, it will establish a rhythm that should prove, over time, beneficial to both parties and aid regular communication methods for feedback and discussion. Deploying the definition of done to increment work will assist the team to set a standard and help GentleApple anticipate the quality of the work. The increment length is extremely useful for assisting the team to determine if the items they have selected from the product backlog are being achieved within the increment timeframe. As the increments cycle through the process and accuracy increases, confidence will also increase with GentleApple stakeholders.

3) *The Sprint Goal* is an important technique to help manage expectations and provide a high-level overview of the incremental work needed to meet the objective, despite this it may be difficult to use given the desire to include stakeholders in the process. GentleApple is expected to have goals they would like to see achieved that are not technically possible, a lack of technical understanding could result in a difficult decision-making process. It will be important for the product owner and development team to articulate why a sprint goal is not possible, whilst also adjusting the goal to ensure it can proceed in a direction that satisfies the business stakeholders. Writing a single sentence that covers the sprint should be easy for an experienced product owner. Once the full team is in agreement the elements can be effectively used to select backlog items that fulfil the goal. Experienced development teams who are familiar with continuous learning through the Scrum cycles means the goal could be easily deployed, making it a well-suited element to the practice of the Scrum framework in the current context. The product owner and development team can easily and effectively make use of previous tasks assigned to sprint goals for measuring the total amount of story points the team can achieve, this will become easier as the project progresses. If an invalidated sprint goal occurs it is likely to disrupt the team and may not be an easy process to work through, the team needs to work with GentleApple to adjust the goal and get back on track as quickly as possible to ensure it does not overly affect the progress of work.

4) *The Sprint Review* meeting acts as a milestone that signifies the end of the sprint and includes a demonstration of the incremental work done. The sprint review structure is well defined by the Scrum framework and requires minimal effort, it should be devoid of detailed slides or formal presentation, instead, the work should form the foundation of the review. In

the case of the GentleApple project and for the benefit of its stakeholders, a few simple slides may be appropriate to introduce the achievements, however, this should not be onerous to create. The regularity of the review means it is easy to schedule virtual attendance by the stakeholders and end-users at GentleApple if necessary, attendance in person should be encouraged to maximise the benefits. The on-screen product demonstration will, on the other hand, be equally as effective from a remote location. The product owner would schedule the event and ensure all relevant stakeholders and users are invited, again this is a simple process and easy to implement into the schedules of both the development team and GentleApple stakeholders, easily fitting into the ways that these disparate teams operate.

6.3 Future Application

There are three main points of note for the GentleApple development, firstly apple processing deploys unique machinery and processes, secondly and crucially, there is a comparatively low computer literacy of employees compared with other industries and thirdly, the work within the scope of development is commonly recognised data entry requirements necessary across business sectors. For example, there are customers, suppliers and stock monitoring, all easily applied across sectors. Considering these factors, the elements of Scrum outlined in this report indicate through the literature, that it is well placed for the development team to use again for future applications. In this context, Scrum is capable of satisfying future outsourced project applications that must consider the traditional success metrics of the 'iron triangle', the criteria of which include time, budget and quality, with quality in Scrum modified to 'scope' or 'requirements' (Hassani-Alaoui et al., 2020). Success metrics in software development include factors that include stakeholder and end-user satisfaction and the meeting of the organisation's goals (Hassani-Alaoui et al., 2020). These are critical considerations for outsourced developments and applying the Scrum technique for the GentleApple project, combined with the learning and ongoing development gained by the team, means that it is well placed to be deployed successfully and to a greater degree of success in future applications.

The benefit of the future use of Scrum is the result of its pillars of transparency, inspection and adaptability (Hassani-Alaoui et al., 2020). This may appear at odds with the Scrum guide that states the artefacts, roles and events are immutable, existing together in

their entirety to be Scrum, with the research indicating that Scrum is capable of being modified to optimise practices to (Hassani-Alaoui et al., 2020):

- Respond to requirements
- Adapt to distributed teams
- Apply the techniques to larger applications
- Adapt to legacy non-agile processes

The research indicates that back to back sprints are consistently followed within Scrum teams, so too is the formation of a product owner with the responsibilities outlined by the position (Hassani-Alaoui et al., 2020). There were several elements identified and not commonly followed by developers, these must not be the case for both the GentleApple project and future developments. Firstly was the concept of protecting teams from distractions, if Scrum is to be followed in future applications the team must show continuous improvement in this area to deliver the anticipated results for clients after the sprint increment (Hassani-Alaoui et al., 2020). Secondly was the use of the backlog as the only source of requirements, this element should not be modified in any future application to ensure the effectiveness of Scrum (Hassani-Alaoui et al., 2020). The final point to note is that teams often do not use the monitoring and sharing tools available to them, including burndown charts, these tools coupled with the product backlog are essential to fulfil the transparency pillar of Scrum and must not be modified in future developments (Hassani-Alaoui et al., 2020).

The research indicates that Scrum is a valuable framework for software development and teams are experiencing higher success rates through the short deployment that reduces risk, eases the identification of architectural or coding errors and getting value to the businesses it contracts with faster. However, it must be remembered that if the pillars are not followed it may lead to problems and missed opportunities when development is outsourced and the company contracting the project, is unfamiliar with Agile methodologies. As a consequence, it is the understanding of the author that this pillar should be practised and improved upon continually to produce quality software, increase stakeholder engagement and provide an MVP in a way that satisfies more traditional success metrics – Scrum is certainly capable of supporting these efforts.

References

- Alhazmi, A., & Huang, S. (2018). A decision support system for sprint planning in Scrum practice. *SoutheastCon 2018*, 1–9.
- Aoufi, A., Schoeman, M., & Turner, N. (2021). How to Outsource Agile Projects Effectively: Suppliers and client advisors need to work closely with client organizations to ensure key enablers are in place to increase success when outsourcing Agile projects. *Research-Technology Management*, 65(1), 59–66.
- BALOG, K. (2019). *AGILE ORGANIZATIONS AND SUSTAINABILITY*.
- Cho, J. (2008). Issues and Challenges of agile software development with SCRUM. *Issues in Information Systems*, 9(2), 188–195.
- Forney, S. J. (2019). *An Agile Success Estimation Framework for Software Projects*. The George Washington University.
- Fowler, F. M. (2019). Scrum Artifacts. In *Navigating Hybrid Scrum Environments* (pp. 55–57). Springer.
- FUIOR, F. (2019). Key elements for the success of the most popular Agile methods. *Romanian Journal of Information Technology and Automatic Control*, 29(4), 7–16.
- Hassani-Alaoui, S., Cameron, A.-F., & Giannelia, T. (2020). “We use scrum, but...”: Agile modifications and project success. *Proceedings of the 53rd Hawaii International Conference on System Sciences*.
- Hernandez, R., Version, D., & Hernandez, R. (2019). Applicability of agile and scrum to product-service systems. *EurOMA Conference*, 1–10.
- High-Performance Teams: Why the “Who” Matters Less | Scrum Alliance | Transforming the World of Work*. (n.d.). Retrieved May 21, 2022, from <https://resources.scrumalliance.org/Article/high-performance-teams-who-matters>
- Hron, M., & Obwegeser, N. (2018). Scrum in practice: An overview of Scrum adaptations. *Proceedings of the 51st Hawaii International Conference on System Sciences*.
- Krancher, O. (2020). Agile Software Development Practices and Success in Outsourced Projects: The Moderating Role of Requirements Risk. In V. Stray, R. Hoda, M. Paasivaara, & P. Kruchten (Eds.), *Agile Processes in Software Engineering and Extreme Programming* (pp. 56–72). Springer International Publishing.
- Kurnia, R., Ferdiana, R., & Wibirama, S. (2018). Software metrics classification for agile scrum process: A literature review. *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 174–179.

- Moe, N. B., Dingsøyr, T., & Dybå, T. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology*, 52(5), 480–491.
- Pereira, J. C., & de FSM Russo, R. (2018). Design thinking integrated in agile software development: A systematic literature review. *Procedia Computer Science*, 138, 775–782.
- Rubin, K. S. (2012). *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley.
- Scrum Team Roles and Responsibilities: Scrum Master and Product Owner*. (n.d.). Software Testing Help. Retrieved May 21, 2022, from <https://www.softwaretestinghelp.com/scrum-roles-responsibilities/>
- Sedano, T., Ralph, P., & Péraire, C. (2019). The product backlog. *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, 200–211.
- Shringi, K. (2020). Ryan Ripley on Making Scrum Work. *IEEE Software*, 38(1), 126–128.
- Sliger, M. (2011, October 22). *Agile project management with Scrum*. <https://www.pmi.org/learning/library/agile-project-management-scrum-6269>
- Young, D. C. (2013). *Software Development Methodologies. White Paper*.
- Zayat, W., & Senvar, O. (2020). Framework study for agile software development via scrum and Kanban. *International Journal of Innovation and Technology Management*, 17(04), 2030002.