

Digital Policy Office

PRACTICE GUIDE FOR AGILE SOFTWARE DEVELOPMENT APPENDIX A TEMPLATES, CHECKLISTS AND SAMPLE DOCUMENTS

[G62a]

Version: 1.4

July 2024

© The Government of the Hong Kong Special Administrative Region
of the People's Republic of China

The contents of this document remain the property of, and may not be reproduced in whole or in part without the expressed permission of the Government of the Hong Kong Special Administrative Region of the People's Republic of China

Amendment History				
Change Number	Revision Description	Pages Affected	Rev. Number	Date
1	As detailed in 1.01 to 1.04		1.1	December 2016
1.01	Add “List of Figures and Tables”	After “Table of Contents” (<i>New</i>)		
1.02	Update the URL of footnote #1	1(b)		
1.03	Update the sub-sections numbering in the sample spreadsheet	7(d), 7.1, 7.2, 7.3, 7.4, 7.5, 7.6		
1.04	Add a new figure - “Project Information” in the section of “A Sample of Spreadsheet for Agile Project Management”	7.1 - Figure 1 added		
2	As detailed in 2.01		1.2	September 2018
2.01	Update to remove criteria related to project criticality for the “Suitability Checklist” section.	3(a) - Table 1 updated		
3	Add the contents of “Design Thinking”	8 – Added Table 15 - added	1.3	May 2019
4	Update Documents for the Establishment of Digital Policy Office (DPO)	Cover page; 1(b) – URL updated	1.4	July 2024

Table of Contents

1	MANIFESTO FOR AGILE SOFTWARE DEVELOPMENT	1
2	TYPES OF AGILE.....	3
3	SUITABILITY CHECKLIST	6
4	A SAMPLE PROJECT SCHEDULE.....	8
5	SAMPLE PROFESSIONAL REQUIREMENTS FOR THE AGILE COACH 11	
6	USER STORIES	12
7	A SAMPLE SPREADSHEET FOR AGILE PROJECT MANAGEMENT.	19
7.1	S1 - Project Information.....	20
7.1.1	Project Schedule	21
7.1.2	Timebox Planning Table.....	22
7.1.3	Burn Down Chart.....	24
7.2	S2 - Prioritised Requirements List (PRL)	25
7.3	S3 - Prioritised Requirements List (with planned tasks).....	26
7.4	S4 - Detailed Tasks List.....	28
7.5	S5 - Story Board	30
7.6	S6 - Task Board	31
8	DESIGN THINKING	32

List of Figures & Tables

Figure 1 -	Project Information	20
Figure 2 -	Project Schedule	21
Figure 3 -	Timebox Planning Table.....	22
Figure 4 -	A Burn Down Chart.....	24
Figure 5 -	Prioritised Requirements List	25
Figure 6 -	Prioritised Requirements List (with planned tasks).....	26
Figure 7 -	Detailed Tasks List	28
Figure 8 -	A Story Board	30
Figure 9 -	A Task Board	31
Table 1 -	Feasibility Criteria	6
Table 2 -	Benefit Criteria	7
Table 3 -	A Sample Project Schedule of Agile Software Development Project	8
Table 4 -	Examples of User Stories.....	12
Table 5 -	Good Examples vs. Bad Examples	13
Table 6 -	Characteristics of a Good User Story	13
Table 7 -	Format of A User Story	14
Table 8 -	Project Schedule Description.....	21
Table 9 -	Timebox Schedule Information	23
Table 10 -	Prioritised Requirements List Description.....	25
Table 11 -	Planned Tasks Description	27
Table 12 -	Detailed Tasks List Description.....	29
Table 13 -	A Story Board Description	30
Table 14 -	A Task Board Description	31
Table 15 -	Five Steps in Design Thinking	32

1 MANIFESTO FOR AGILE SOFTWARE DEVELOPMENT

(a) In 2001, some Agile practitioners introduced four main values for developing software using Agile under the "Manifesto for Agile Software Development". The values are listed as follows:

1) Individuals and Interactions over Processes and Tools

Self-organisation of a project team, and motivation and collaboration of team members are more important than just following the processes and tools.

2) Working Software over Comprehensive Documentation

Working software will be more useful in reflecting the progress of work than just presenting documents to clients.

3) Customer Collaboration over Contract Negotiation

Customer and stakeholders are encouraged to continuously collaborate with project team members throughout the project rather than negotiate according to the terms and conditions in the contract.

4) Responding to Change over Following a Plan

Agile focuses on quick and immediate responses to changes rather than strictly following a plan.

(b) To supplement the Manifesto, there were twelve Agile Principles developed in 2001 to elaborate the Agile approach. The following are the twelve Agile Principles¹:

- 1) Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- 2) Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- 3) Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- 4) Business people and programmers must work together daily throughout the project.
- 5) Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- 6) The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- 7) Working software is the primary measure of progress.

¹ Source: <https://agilemanifesto.org/iso/en/principles.html>

- 8) Agile processes promote sustainable development. The sponsors, programmers, and users should be able to maintain a constant pace indefinitely.
- 9) Continuous attention to technical excellence and good design enhances agility.
- 10) Simplicity--the art of maximising the amount of work not done--is essential².
- 11) The best architectures, requirements, and designs emerge from self-organising teams.
- 12) At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

² It recommends doing just enough work in order to reduce the wasted effort.

2 TYPES OF AGILE

- (a) The Agile practices currently available in the market may be classified into two major approaches:
- i) **Disciplined Agile Approach**
The Disciplined Agile Approach is a structural approach which covers the entire System Development Life Cycle (SDLC) from project initiation till system live-run. It usually incorporates a comprehensive set of structured processes and Agile practices that work for the entire system development life cycle. Two examples of Agile methods are the “Dynamic Systems Development Method (DSDM)” and the “Open Unified Process (OpenUP)”.
 - ii) **Core Agile Approach**
The Core Agile Approach such as “Scrum” and “Extreme Programming (XP)” is a less structural approach covering only a portion of the SDLC. It emphasises the technical aspects of system delivery mechanism and only addresses the System Implementation Phase of the SDLC.
- (c) Each Agile method has its own characteristics and merits to suit the needs of different organisations.
- (d) To adopt Agile for implementation of IT systems in the Government, a mixed approach is used to select and consolidate the best practices from different Agile methods that are applicable to the Government.
- (e) More information about the above examples of Agile methods is provided below.

DSDM

Dynamic Systems Development Method (DSDM) is an Agile project delivery framework that adopts an iterative and incremental approach for system development. The most recent version, DSDM Atern, was launched in 2007. DSDM was originally aimed to provide discipline to the Rapid Application Development (RAD), but it gradually became a more generic approach covering project management and solution delivery. DSDM considers that time and cost for a project are fixed while only requirements are allowed to be changed.

The following are some core techniques used in DSDM:

- **Timeboxing** - A timebox is an interval usually no longer than 6 weeks. Each timebox contains several tasks which are pre-determined at the project planning stage, and are required to deliver a working software at the end;
- **MoSCoW Rules** - The MoSCoW prioritisation method produces a "Prioritised Requirements List" (PRL) in which the requirements are arranged in the order of "Must have", "Should have", "Could have" and "Won't have" to adjust the system functions to be delivered to meet the project time and other constraints;

- **Prototyping** - Prototypes are used to implement critical functionality to discover any difficulties in early stage of development, and to collect user feedback in a more interactive way; and
- **Facilitated Workshop** - The idea of workshop is implemented to facilitate communication and establish collaboration between users and project team.

OpenUP

The Unified Process (UP) is a common iterative and incremental system development process framework which consists of four main phases. They are inception, elaboration, construction and transition phases. These phases can be further divided into a series of timeboxed iterations.

The Open Unified Process (OpenUP) is an open source process framework developed to make it easier to adopt the core of the Unified Process. OpenUP preserves the characteristics of UP including incremental development, use of cases and scenarios as well as architecture-centric development approach.

Four core principles of OpenUP are stated below:

- **Collaborate to align interests and share understanding** - Foster a healthy team environment, enabling collaboration between users and project team and development of a shared understanding of the project;
- **Evolve to continuously obtain feedback and improve** - Allow project team to get early and continuous feedback from users, and demonstrate incremental value to them;
- **Balance competing priorities to maximise stakeholder value** - Allow project team and users to develop a solution that maximises stakeholder benefits, yet is still compliant with project constraints; and
- **Focus on articulating the architecture** - Allow project team to focus on architecture to support the development and to minimise project risks.

Scrum

Scrum is a popular Agile method using iterative and incremental approach for system development. It has a set of roles and practices. One major role is the “Scrum Master”, who acts as a coach or facilitator to guide the project team and users to ensure that the Scrum process is used properly and keep the team focus on the tasks at hand. For example, the Scrum Master will help the project team maximise the productivity to deliver the functions quickly, and help the “Product Owner” who is the project owner or the representative of stakeholders maintain the prioritised list of user requirements which is usually named as "Product Backlog".

A "sprint" is the basic unit of development in Scrum, and is often referred to as a timebox covering a constant length of period of time say for example, 2 to 4 weeks. The duration of the sprint is often determined by the Scrum Master. Each sprint starts with a scrum planning meeting to let users and project team identify the details of tasks to be done during the sprint and define the criteria for accepting the work done. At the end of the sprint, a review meeting will be held to review the progress and to approve or reject the work done according to the pre-defined criteria.

XP

Extreme Programming (XP) is a system development method which aims at rapid development of high quality and productivity system at lower cost and faster responsiveness of the programmers to changing customer requirements. XP teams often build a system in 2-week iterations, delivering working component at the end of each iteration based on user stories that are written by users stating the requirements in a layman way.








XP is based on values of communication, simplicity, feedback, and respect and courage, which are briefly described below:

- **Communication** - encourages constant and close communication between users and programmers;
- **Simplicity** - encourages development to start from a simple, planned solution and then add extra functionality;
- **Feedback** - provides immediate feedback from the unit testing performed by system if an automated testing tool is used, and the functional testing performed by user;
- **Respect** - builds up team members' respect for their unique contributions by every small success; and
- **Courage** - refactors code when necessary, easily evolves the code and makes small changes on the code to support new requirements, and thus improve the overall quality.

3 SUITABILITY CHECKLIST

- (a) Table 1 illustrates how the criteria are used to consider the feasibility of Agile. Agile is considered feasible if all the criteria under this category are skewed to the right-hand side. However, the existence of negative responses may not necessarily indicate non-feasibility. If additional cautions are taken and potential impacts on the Agile project execution are properly managed, Agile is still feasible for the project.





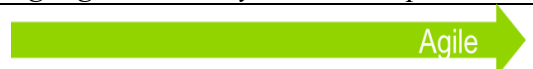
Table 1 - Feasibility Criteria

Feasibility Criteria			
Project aspect			
1. Project team size	Large		Small
<i>Agile prefers small teams, or the large team can be divided into smaller teams.</i>			
2 Team location	Remote		Co-located
<i>Agile prefers project teams to sit together physically or at least virtual communication tools are used for geographically distributed teams.</i>			
3. External dependency	High		Low
<i>It is desirable that systems have low dependencies with other systems or parties (e.g. do not rely on information provided by systems from other departments). Nevertheless, Agile can still be used if the impacts from external dependencies are able to be foreseen and controllable.</i>			
People aspect			
4. Staff knowledge and skills in Agile	Low		High
<i>Staff knowledge and skills on the use of Agile methodology may affect the smoothness of Agile project delivery.</i>			
5. Staff stability	Low		High
<i>Maintain staff stability by avoiding staff turnover or ensuring effective hand-over of tasks.</i>			
6. User involvement & interaction	Seldom		Frequent
<i>Agile requires that full-time user representatives are dedicated to the project with delegated decision power, and are willing to participate, communicate and make authorised decisions.</i>			
Technology aspect			
7. Tools availability for solution	Low		High

Agile requires appropriate development tools to allow for iterative development and testing, demonstrable work products and control of versions. Code construction should be rapid and easily testable with automated build and testing tools.

- (b) The following table indicates the characteristics of a project that could benefit the most from Agile.

Table 2 - Benefit Criteria

Benefit criteria			
1. Requirement volatility	Stable		Volatile
<p>Characteristic: Requirements are volatile, i.e. there is a high possibility of changes in user requirements, or high uncertainty in user requirements.</p> <p>Benefit: The use of Agile can improve the flexibility of implementing requirement changes because low-level requirements can be swapped rapidly.</p>			
2. Product novelty and innovation	Less		More
<p>Characteristic: The deliverables are new to the market or to the Government (e.g. Web 2.0, HTML5).</p> <p>Benefit: As innovative product usually evolves along the implementation, detailed up-front design is not required. Whereas, the Agile approach on design such as modelling and prototyping techniques is more appropriate. Therefore, Agile helps reduce SA&D lead time.</p>			
3. Time to delivery	Slow		Fast
<p>Characteristic: On-time / shorter time to delivery is required or highly desired.</p> <p>Benefit: The nature of timeboxing approach and frequent release capability of Agile lead to timely delivery of the product.</p>			
4. Demand for visibility	Low		High
<p>Characteristic: Users demand for early preview of the product such as user interface design and features as well as the progress of the project.</p> <p>Benefit: Better project tracking and monitoring can be achieved with timeboxing approach, which can help bring higher visibility to both the product and project.</p>			
5. Extensibility	Low		High
<p>Characteristic: Systems that can be built as components or require high extensibility (e.g. a core system with incremental add-on components or features).</p> <p>Benefit: Business benefits will be realised early by delivering self-contained components with more frequent and predictable software releases using Agile.</p>			

4 A SAMPLE PROJECT SCHEDULE

The following table shows a sample project schedule with various deliverables. A project team should suitably adjust the schedule, milestones and deliverables according to the project nature and needs. If there are multiple releases to be rolled out to production, some milestones (e.g. 5 to 9 as shown in the following table) may be repeated in each release according to the project needs.

Table 3 - A Sample Project Schedule of Agile Software Development Project

Milestone	End of Date	Deliverables
1. Project Initiation	1 st Month	<ul style="list-style-type: none"> • Project Initiation Document (if project is outsourced)
2. System Design < In agile approach, duration for system design is normally shorter than the waterfall approach.>	2 nd Month	<ul style="list-style-type: none"> • Requirement Specification <ul style="list-style-type: none"> ➤ User Requirements Document <ul style="list-style-type: none"> ◆ Prioritised Requirements List ◆ Non-functional Requirements ◆ Future Business Process ➤ Technical Requirements • System Specification <ul style="list-style-type: none"> ➤ Functional Specification <ul style="list-style-type: none"> ◆ High-level Required System Overview ◆ High-level Functional Definition ➤ Architecture Design ➤ System Design <ul style="list-style-type: none"> ◆ High-level Logical Data Model (e.g. Entity Relationship Diagram, Class Diagram) ◆ User Experience (using prototyping techniques) • Technical System Option
3. Site Preparation and Cabling	4 th month	<ul style="list-style-type: none"> • Site Specification • Accepted site
4. Procurement & Installation of Hardware and Software	4 th month	<ul style="list-style-type: none"> • Procurement Plan • Procurement Specification and/or Procurement List • Hardware and Software procured, installed and accepted
5. Program Development, Unit Tests and Integration Tests	4 th Month	<ul style="list-style-type: none"> • Revised Prioritised Requirements List • Revised Prototypes/Models • Test Plan and Test Result

Milestone	End of Date	Deliverables
6. User Acceptance <i>< In agile approach, duration for user acceptance is normally shorter than waterfall approach ></i>	5 th Month	<ul style="list-style-type: none"> • Acceptance Test Plan for the overall system, Specifications and Results • "Passing tests" software components
7. Training	6 th Month	<ul style="list-style-type: none"> • Training Plan • Training Manual • Training Courses • Trained Users
8. Prepare System Documentation <i>< In agile approach, a minimum set of documentation is required, e.g. Application Operation Manual and Application User Manual. However, System Manual, Program Manual, Data Manual and Computer Operation Procedures Manual will be produced for on-going maintenance and operation support as necessary.></i>	6 th Month	<ul style="list-style-type: none"> • Application Operation Manual • Application User Manual <p><i><Below are for mission critical systems ></i></p> <ul style="list-style-type: none"> • Business Impact Analysis <i><if not yet in place or the reviewed one if already in place>;</i> • Business Continuity Plan which includes a Disaster Recovery Plan
9. System Installation & Production	7 th Month	<ul style="list-style-type: none"> • System Installation Plan • H/W and S/W modules in integration • System Installation Report • System roll-out and in production use
10. Nursing	10 th Month	<ul style="list-style-type: none"> • System Nursing & Maintenance; • Hand-over Plan

Milestone	End of Date	Deliverables
11. Disaster Recovery Drill <if required>	11 th Month	• Disaster Recovery Drill Test Report
12. Project Closure	11 th Month	• Project Evaluation Report

5 SAMPLE PROFESSIONAL REQUIREMENTS FOR THE AGILE COACH

In acquiring the Agile coaching/mentoring services, it is recommended that the Agile Coach has the following areas of experience and qualification:

- (a) Possessed *<xxx years>* of working experience in Information Technology (IT);
- (b) Delivered *<xxx>* IT Projects* using agile method (project management or system development services, excluding coaching services) with each project having a duration of *<xxx>*;
- (c) Provided coaching services to users and/or project teams in using agile method for *<xxx>* IT Projects* with each project having a duration of *<xxxx>*; and
- (d) Awarded *<xxx>* of the following Agile certifications:
 - Certified Scrum Coach (CSC) by Scrum Alliance
 - Certified Scrum Trainer (CST) by Scrum Alliance
 - PMI Agile Certified Practitioner (PMI-ACP) by Project Management Institute

* IT Project refers to a project to provide IT system implementation service(s)

6 USER STORIES

- (a) User stories act as requirements on the system, and are written in layman’s terms which are not technical. This facilitates verbal communication between end users and project teams.
- (b) The content of a user story should include the following information:
- Role of the user
 - Function/features the user wants to have
 - Benefits/achievements/reasons to have the function/feature
- (c) For clarity in communication, it is suggested that the user story is presented in a structured format as follows:
- As a/an **<user role>**...
- I want to **<goal>**...
- so that **<reason>**.
- (d) Alternatively, user story can be written in free format to suit the project needs or user preferences. An example of user story written in free format is shown below:
- “A login password should be provided to each user to allow them to protect his/her own user account from unauthorised access.”
- (e) There is no strict rule on how a user story should be written as far as the requirements presented are clear and capable of providing information necessary for project team to start the timebox development.
- (f) The following tables show several examples of user stories. The words that are written in “*bold italic*” are sample contents for reference.

Table 4 - Examples of User Stories

No.	As a/an	I want to...	So that...
1	<i>user</i>	<i>bookmark the current page</i>	<i>I can quickly find the same page next time.</i>
2	<i>user</i>	<i>change my login password</i>	<i>I can better protect my account.</i>
3	<i>job seeker</i>	<i>search the job vacancies by a keyword</i>	<i>I can save time finding target job vacancies.</i>
4	<i>student</i>	<i>submit my sick leave application online</i>	<i>I can save time to go to the Student Affairs Counter to submit the sick leave application form in person.</i>
5	<i>Manager</i>	<i>view my projects status in one page</i>	<i>I can keep track of the projects at a glance.</i>

- (g) The followings are some good and bad examples of user stories.

Table 5 - Good Examples vs. Bad Examples

Good Example	Bad Example
<i>As a Supplies Officer, I want to access the product marketing information so that I can make a purchasing recommendation.</i>	<i>As a user, I want to access the stored information so that I can learn more.</i>
<i>As a conference attendee, I want to register online so that I can save time on paperwork.</i>	<i>As a conference attendee, I want to click on the "Register" button next to the conference name to register online so that I can save time on paperwork.</i>

- (h) A well-written user story should possess the following six common characteristics, with an acronym "INVEST" for easy reference.

Table 6 - Characteristics of a Good User Story

Characteristics	Description
<u>I</u> ndependent	User stories should be self-contained, independent of other user stories.
<u>N</u> egotiable	User stories should be allowed for change and rewritten until they start to be implemented in a timebox.
<u>V</u> aluable	Each user story should deliver a business value to the user.
<u>E</u> stimable	Each user story should be capable of being estimated of its size i.e. required effort.
<u>S</u> mall	User stories should be small enough to fit within a timebox.
<u>T</u> estable	A user story needs to provide information that is necessary for testing i.e. to make the development of tests possible.

- (i) Each user story should have a story title, story content, acceptance criteria, a submitted person, the submission date and also the last modified date. The acceptance criteria should be written in simple language, and should not contain any technical terms.

Table 7 - Format of A User Story

Item	Description
Story Title:	Title of a user story
Story Content:	The content of a user story
Acceptance Criteria:	A list of acceptance criteria for the user story
Submitted By:	Name and post/rank of a person who creates the user story
Submitted On:	The submission date
Last Modified:	The last modified date

(j) More examples on user stories are listed below:

User Story Example 1	
Story Title:	<i>Computer Items Lending</i>
Story Content:	<i>As a Computer Centre Administrator, I want to process the lending of computer items so that the computer items lent are properly recorded and a maximum of five items can be borrowed by one authorised staff at a point of time.</i>
Acceptance Criteria:	<ul style="list-style-type: none"> <i>(a) Computer items lent can be successfully recorded.</i> <i>(b) Support manual input either by keyboard or by a barcode scanner to record Staff ID and the items' barcodes.</i> <i>(c) An item can only be borrowed by one person at a given time.</i> <i>(d) Alert if the item has already been borrowed by another person at a given time.</i> <i>(e) Alert if the staff ID is invalid e.g. retired, resigned, transferred out to other B/D, etc.</i> <i>(f) Alert if the staff has borrowed five items already.</i> <i>(g) A maximum 5 items can be borrowed at a given time after the input of the staff ID.</i> <i>(h) The Computer Centre Administrator does not need to hit any keystroke when using a barcode scanner unless an alert is prompted or the lending request is to be cancelled.</i>
Submitted By:	<i>Mr. AAA, Computer Centre Administrator</i>
Submitted On:	<i>01.08.2014</i>
Last Modified:	<i>05.08.2014</i>

User Story Example 2	
Story Title:	<i>Login</i>
Story Content:	<i>As a registered user, I want my login to be validated so that I can obtain the required access rights to perform various user functions.</i>
Acceptance Criteria:	<i>(a) Format of the user login ID should be of the same format as the Departmental Portal ID, i.e. "[name].[dept/bureau]" to login. Login hints (such as format of the login ID) should be displayed for user, special characters should not be allowed. (b) Login ID in other format will be rejected and display an error message showing "Invalid user login ID". (c) Login ID should be a valid Departmental Portal ID and should be a registered user in the system. Otherwise, an error message "You are not authorised to access the system since you are not a registered user."</i>
Submitted By:	<i>Mr. BBB, User of Division B</i>
Submitted On:	<i>01.08.2014</i>
Last Modified:	<i>13.08.2014</i>

User Story Example 3	
Story Title:	<i>Reset Password</i>
Story Content:	<i>As a registered user, I wish I can recover my login ID and/or password so that I can login again if my login ID was locked by the system.</i>
Acceptance Criteria:	<i>(a) A button to allow user to submit a request to recover login password. (b) An email will be automatically sent to the User Administrator by the system upon receipt of a password reset request. (c) The administrator can then unlock the user account and randomly re-generate a new password from the system and send it to the registered email address of the user. In the email, the login information and new password is listed, and a message should be stated to request user to reset the password immediately once login the system. (d) The login password generated by the system should be of 8 alphanumeric characters long.</i>
Submitted By:	<i>Mr. BBB, User of Division B</i>
Submitted On:	<i>01.08.2014</i>
Last Modified:	<i>13.08.2014</i>

User Story Example 4	
Story Title:	<i>Search Computer Items</i>
Story Content:	<i>As a user, I want to search the computer items by selecting an item type and/or specifying a keyword for any item with description that contains the keyword. After searching, a list of search results containing the matched item types and/or item descriptions, and their corresponding available quantities and location should be displayed. I want to have the search results displayed in a scrollable list for ease of navigation. When there are too many searched items, I accept go to the next page to continue the viewing of search result.</i>
Acceptance Criteria:	<i>(a) Search result can be displayed successfully. (b) Paging can be performed successfully for large volume of search results. (c) Item details can be displayed successfully.</i>
Submitted By:	<i>Mr. CCC, User of Division C</i>
Submitted On:	<i>01.08.2014</i>
Last Modified:	<i>23.08.2014</i>

User Story Example 5	
Story Title:	<i>Statistical Report</i>
Story Content:	<i>As a Computer Centre Administrator, I want to have a statistical report on the number of transactions on lending, reservation and renewal of computer items for a specific date range.</i>
Acceptance Criteria:	<i>(a) The number of lending, reservations and renewals are displayed correctly on the report. (b) An online function for printing the report should be provided.</i>
Submitted By:	<i>Mr. AAA, Computer Centre Administrator</i>
Submitted On:	<i>01.08.2014</i>
Last Modified:	<i>22.08.2014</i>

User Story Example 6	
Story Title:	<i>Assign case number</i>
Story Content:	<i>As a Supplies Officer, I want to assign an approval case number to the current Inventory Holder who has issued a request for the disposal of computer items so that the disposal process can be completed following the Stores and Procurement Regulations.</i>
Acceptance Criteria:	<i>(a) A unique case number is assigned to each disposal request. (b) Upon receipt of a valid disposal request, trigger the flow of disposal process, otherwise reject the invalid disposal request. (c) Notify all related parties for the valid disposal request.</i>
Submitted By:	<i>Mr. SSS, Supplies Officer</i>
Submitted On:	<i>01.08.2014</i>
Last Modified:	<i>22.08.2014</i>

User Story Example 7	
Story Title:	<i>History Enquiry</i>
Story Content:	<i>As a Supplies Officer, I want to enquire the history of transactions so that I can view all past transactions for a specific period of time.</i>
Acceptance Criteria:	<i>(a) Able to enquire the following transaction history for a specific period of time:</i> <ul style="list-style-type: none"><i>● Update of Computer Items;</i><i>● Issue of Computer Items;</i><i>● Hand-Over and Take- Over of Computer Items;</i><i>● Transfer of Computer Items;</i><i>● Trade-in of Computer Items;</i><i>● Write-off of Computer Items; and</i><i>● Disposal of Computer Items.</i>
Submitted By:	<i>Mr. SSS, Supplies Officer</i>
Submitted On:	<i>01.08.2014</i>
Last Modified:	<i>23.08.2014</i>

User Story Example 8	
Story Title:	<i>Audit Trail Report</i>
Story Content:	<i>As a Supplies Officer, I want to prepare an audit trail report so that I can trace the activities related to the system.</i>
Acceptance Criteria:	<i>(a) Keep a log of user access to the login function and login. (b) Keep a log of user retrieval and updating of computer items information on the system. (c) Keep a log of data changed (before and after images) for all data updating.</i>
Submitted By:	<i>Mr. SSS, Supplies Officer</i>
Submitted On:	<i>01.08.2014</i>
Last Modified:	<i>23.08.2014</i>

User Story Example 9	
Story Title:	<i>Duplicate Computer Item</i>
Story Content:	<i>As a Supplies Officer, I want to copy details of a computer item to create another new computer item so that I can duplicate similar item details to save input time.</i>
Acceptance Criteria:	<i>(a) Create a new item which is copied from another existing item with the following information prefilled: item type, item description, maintenance/warranty period, location, quantities, unit price, total price, etc.</i>
Submitted By:	<i>Mr. SSS, Supplies Officer</i>
Submitted On:	<i>01.08.2014</i>
Last Modified:	<i>23.08.2014</i>

7 A SAMPLE SPREADSHEET FOR AGILE PROJECT MANAGEMENT

- (a) Besides the Agile software tools, the project team may consider to use a spreadsheet or a worksheet to monitor the project information and progress.
- (b) A spreadsheet helps maintain the user stories, PRL and tasks list and generate the burn down charts, story boards and task boards.
- (c) A spreadsheet template is prepared using some of the examples of user stories in *Section 6*.
- (d) The spreadsheet contains the following six worksheets covering:
 - i) S1 - Project information
 - ii) S2 - PRL
 - iii) S3 - PRL (with planned task)
 - iv) S4 - Detailed Tasks list
 - v) S5 - Story board
 - vi) S6 - Task board
- (e) The sample contents which are in form of tables or charts are extracted below. Input values for the sample contents are marked in “bold italic” and should be replaced by project-specific information to suit specific project needs. Upon input of values, fields shown in grey colour will be updated automatically by some pre-defined rules or formulas set in the worksheet.

7.1 S1 - PROJECT INFORMATION

This worksheet includes the basic project schedule, the timebox planning table and the burn down chart.

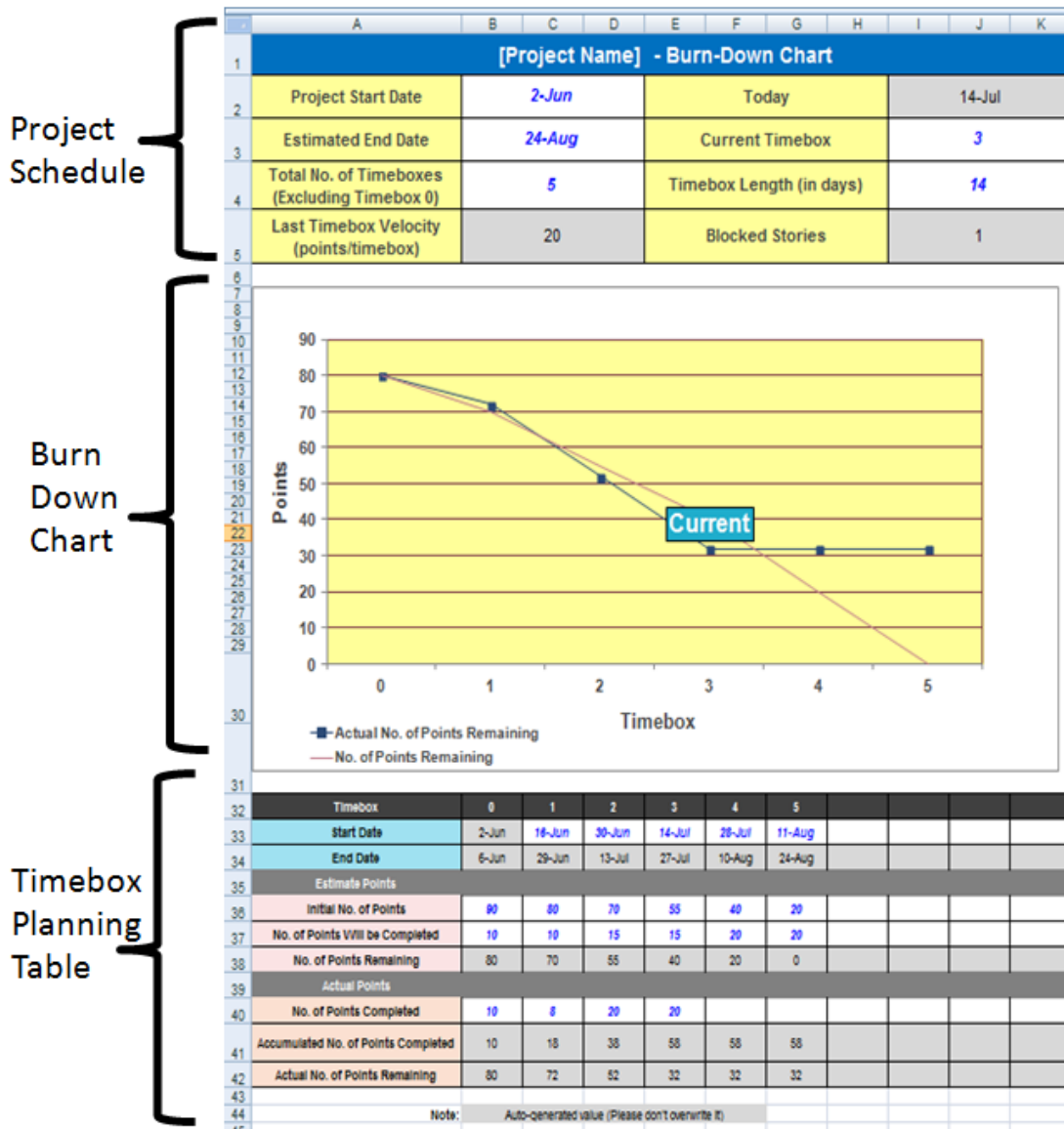


Figure 1 - Project Information

7.1.1 Project Schedule

The project team should fill in the project schedule in Figure 2.

	A	B	C	D	E	F	G	H	I	J	K
1	[Project Name] - Burn-Down Chart										
2	Project Start Date	2-Jun			Today			14-Jul			
3	Estimated End Date	24-Aug			Current Timebox			3			
4	Total No. of Timeboxes (Excluding Timebox 0)	5			Timebox Length (in days)			14			
5	Last Timebox Velocity (points/timebox)	20			Blocked Stories			1			

Figure 2 - Project Schedule

Table 8 - Project Schedule Description

Field Name	Description
Project Name	Fill in the name of the project.
Project Start Date*	Fill in project start date.
Estimated End Date*	Fill in project estimated end date.
Total No. of Timeboxes (Excluding the Timebox 0)	Fill in the total number of planned timeboxes for the project.
Last Timebox Velocity (points/timebox)	An auto-generated field showing the velocity of last timebox.
Today*	An auto-generated field with today’s system date.
Current Timebox	Fill in the current timebox number e.g. 3 stands for the third timebox.
Timebox Length (in days)	Fill in a fixed length for all timeboxes, i.e. a fixed number of time interval in calendar days, normally ranging from 7 days (1 week) to 28 days (4 weeks).
Blocked Stories	An auto-generated field with the number of blocked user stories when it was selected as “Blocked” status at the PRL (if any).

*Note: The format of all date fields displayed can be flexibly set, e.g. dd-mon-yy, dd-mon-yyyy, dd-mm-yyyy, etc.

7.1.2 Timebox Planning Table

The project start date in Figure 2 will be automatically copied as the Start Date of Timebox 0 in Figure 3 in the worksheet. Project team should fill in the start date of all timeboxes throughout the timebox period until all timeboxes are completed.

32	Timebox	0	1	2	3	4	5
33	Start Date	2-Jun	16-Jun	30-Jun	14-Jul	28-Jul	11-Aug
34	End Date	6-Jun	29-Jun	13-Jul	27-Jul	10-Aug	24-Aug
35	Estimate Points						
36	Initial No. of Points	90	80	70	55	40	20
37	No. of Points Will be Completed	10	10	15	15	20	20
38	No. of Points Remaining	80	70	55	40	20	0
39	Actual Points						
40	No. of Points Completed	10	8	20	20		
41	Accumulated No. of Points Completed	10	18	38	58	58	58
42	Actual No. of Points Remaining	80	72	52	32	32	32
43							
44	Note:	Auto-generated value (Please don't overwrite it)					
45							

Figure 3 - Timebox Planning Table

*Note: This is an example of using story points to represent the effort of tasks. Project teams can also use man-days instead of story points in estimating the staff effort. Story points or man-days are simply two different units of measurements used to express how much staff effort is required to complete the planned user stories in a timebox. The calculation formula used in the spreadsheet will remain unchanged.

Table 9 - Timebox Schedule Information

Field Name	Description
Timebox	An auto-generated field to indicate the sequence number of each timebox.
Start Date	Fill in the start date of each timebox (except the first one which was copied from the project start date provided in Figure 2).
End Date	An auto-generated field with the end date of each timebox.
Estimated Points	
Initial No. of Points	Fill in the total no. of points planned in each timebox, which is roughly estimated at the Release Planning meeting and can be revised before the timebox starts.
No. of Points Will be Completed	Fill in the estimated no. of points that will be completed in each timebox.
No. of Points Remaining	Auto-generated fields showing the estimated no. of points remained (i.e. Initial No. of Points minus No. of Points will be Completed) in each timebox.
Actual Points	
No. of Points Completed	Fill in the actual no. of points completed (i.e. done) for the timebox.
Accumulated No. of Points Completed	Auto-generated fields showing the accumulated total no. of points that were completed from timebox 0 to the end of the current timebox.
Actual No. of Points Remaining	Auto-generated fields showing the accumulated total no. of points left (i.e. Initial No. of Points minus Accumulated No. of Points Completed) in each timebox.

7.1.3 Burn Down Chart

- (a) A burn down chart shown in Figure 4 will be auto-generated based on the information in Figure 2 and Figure 3.

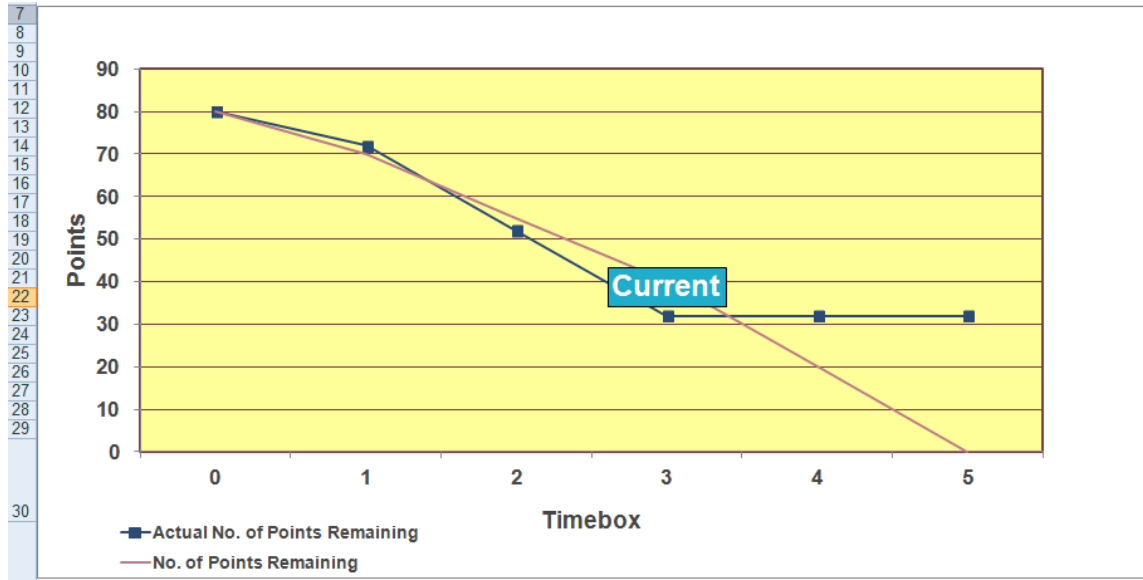


Figure 4 - A Burn Down Chart

- (b) The chart shows two running down lines for the planned estimated points (planned line) and the actual completed points (actual line) respectively. The amount of work left (i.e. the work that needs to be completed for the project) is shown on the vertical axis, while the project timeline is shown along the horizontal x-axis.
- (c) The actual line and the planned line initially start at the same point. As time progresses, the actual line fluctuates above or below the planned line depending on how effective the team is. If the actual line is above the planned line, it means that there is more work left than originally estimated and the project is behind schedule. On the other hand, if the actual line is below the planned line, it means that there is less work left than originally estimated and the project is thus ahead of schedule.
- (d) Besides the above tables for project management, BA should help users fill in a PRL based on the defined user stories. A PRL contains the user story name, acceptance criteria, priority (or business value) and assumption. A sample PRL is shown in Figure 4.

7.2 S2 - PRIORITISED REQUIREMENTS LIST (PRL)

BA should copy the user stories from the PRL which was confirmed by users at the Release Planning meeting, into the worksheet. This list should be continuously updated by the BA throughout the timebox period if there are any changes in the user stories. The project team will make use of the information to update the corresponding planning tasks after discussion in the timebox planning meeting.

[Project Name] - Prioritized Requirement List				
Story Title	Story Content	Acceptance Criteria	Priority	Assumptions
Computer Items Lending	As a Computer Centre Administrator, I want to process the lending of computer items so that the computer items lent are properly recorded and a maximum of five items can be borrowed by one authorised staff at a point of time.	(a) Computer items lent can be successfully recorded. (b) Support manual input either by keyboard or by a barcode scanner to record Staff ID and the items' barcodes. (c) An item can only be borrowed by one person at a given time.	Must-have	No
Login	As a registered user, I want my login to be validated so that I can obtain the required access rights to perform various user functions.	(a) Format of the user login ID should be of the same format as the Departmental Portal ID, i.e. "[name].[dept/bureau]" to login. Login hints (such as format of the login ID) should be displayed for user, special characters should not be allowed.	Must-have	No
Reset Password	As a registered user, I wish I can recover my login ID and/or password so that I can login again if my login ID was locked by the system.	(a) A button to allow user to submit a request to recover login password. (b) An email will be automatically sent to the user administrator by the system upon receipt of a password reset request.	Must-have	No

Figure 5 - Prioritised Requirements List

Table 10 - Prioritised Requirements List Description

Column Name	Description
Story Title	The name of a user story
Story Content	The content of a user story
Acceptance Criteria	The detailed acceptance criteria for the user story such as business rules, validation checks, processing, input to/output from the user story, and so on.
Priority	The value of relative importance of the user story, which can be assigned with values such as "Must-have", "Should-have", "Could-have", "Won't-have", or with relative numbers, say 1 to 5.
Assumptions	Assumptions or remarks of the user story.

7.3 S3 - PRIORITISED REQUIREMENTS LIST (WITH PLANNED TASKS)

During the timebox period, the project team should update information related to the planned tasks for the user stories such as the planned story points or man-days estimated for each user story, the assigned programmer name and project status in the worksheet. This list should be continuously updated by the project team throughout the timebox period.

[Project Name] - Prioritised Requirement List												
Story ID	Time box	Story Title	Acceptance Criteria	Priority	Assumptions	Planned story points	Assigned To	Blocked	External Dependency	Status	Creation Date	Last Date Updated
1	1	Computer Items Lending	(a) Computer items lent can be successfully recorded. (b) Support manual input either by keyboard or by a barcode scanner to record Staff ID and the items'	Must-have	No	5	Ada	Yes	N/A	Testing	16-May	16-May
2	1	Login	(a) Format of the user login ID should be of the same format as the Departmental Portal ID, i.e. "[name].[dept/bureau]" to login. Login hints (such as format of the	Must-have	No	5	Benny	No	N/A	In Progress	16-May	16-May
3	2	Reset Password	(a) A button to allow user to submit a request to recover login password. (b) An email will be	Must-have	No	10	Benny	No	N/A	To Do	16-May	16-May

Figure 6 - Prioritised Requirements List (with planned tasks)

Table 11 - Planned Tasks Description

Column Name	Description
Story ID	A unique sequence ID assigned to each user story.
Timebox	A timebox number assigned to each user story during the Release Planning meeting.
Story Title*	The name of a user story
Acceptance Criteria*	The detailed acceptance criteria for the user story such as business rules, validation checks, processing, input to/output from the user story, and so on.
Priority*	The value of relative importance of the user story which could be assigned with values such as “Must-have”, “Should-have”, “Could-have”, “Won’t-have”, or with relative numbers, say 1 to 5.
Assumptions*	Assumptions or remarks of the user story.
Planned number of Story points (man-days)	The total estimated effort required in terms of story points (or man-days) for the user story.
Assigned To	The name of the project team member i.e. the programmer/SA who has been assigned to implement this user story.
Blocked	“Yes” if the user story was finally identified as blocked i.e. could not be done at the end of the timebox. Default value is “No”.
External Dependency	The external dependency affecting the progress of the user story. Default value is “N/A”. (e.g. the development task of a planned function to capture a scanned image of a document could only be done after the bar code scanner has been delivered.)
Status	The progress of the user story such as “Defined”, “To Do”, “In Progress”, “Tested”, “Done”, “Removed”.
Creation Date	The creation date of the user story
Last Updated Date	The last updated date of the user story

**Note: The columns are initially copied from the PRL shown in Figure 5 and project team will have to update the user stories in the list during the timebox period if there are any changes.*

7.4 S4 - DETAILED TASKS LIST

Each user story will be broken down into detailed tasks for implementation. The project team should fill in the tasks list and maintain the progress of each individual task as well as the overall progress of the user story in the PRL in Figure 5 accordingly.

[Project Name] - Task List											
Time box	Task ID	Story ID	Story Title	Task	Assigned To	Planned no. of man-days (Points)	Blocked	External Dependency	Status	Creation Date	Last Date Updated
1	1	1	Computer Items Lending	Layout design	Ada	2	No	N/A	Testing	16-May	16-May
1	2	1	Computer Items Lending	Backend logic on lending	Ada	2	No	N/A	Testing	16-May	16-May
1	3	1	Computer Items Lending	Validation rules creation	Ada	1	No	N/A	Testing	16-May	16-May
1	4	2	Login	Form design and creation	Benny	2	No	N/A	In Progress	16-May	16-May
1	5	2	Login	SQL creation	Benny	3	No	N/A	In Progress	16-May	16-May
2	6	3	Reset Password	Layout design	Benny	2	No	N/A	To Do	16-May	16-May
2	7	3	Reset Password	SQL enquiry	Benny	5	No	N/A	To Do	16-May	16-May
2	8	3	Reset Password	Backend logic on reset and return password	Benny	3	No	N/A	To Do	16-May	16-May

Figure 7 - Detailed Tasks List

Table 12 - Detailed Tasks List Description

Column Name	Description
Timebox	A timebox number assigned to each user story during the Release Planning meeting.
Task ID	A unique sequence ID assigned to each task
Story ID	A unique sequence ID assigned to each user story.
Story Title	The name of the user story.
Task	The detailed task description with detailed requirements and acceptance criteria.
Assigned To	The name of the project team member i.e. the programmer/SA who has been assigned to implement the detailed task of the user story. Normally, all tasks of a user story will be assigned to the same person.
Planned number of man-days (Points)	The estimated effort required for the detailed task in terms of man-days (or story points).
Blocked	“Yes” if the detailed task of the user story was identified as blocked, i.e. could not be done within the schedule. Default value is “No”.
External Dependency	The external dependency affecting the progress of the user story. Default value is “N/A”. (e.g. the development task of a planned function to capture a scanned image of a document could only be done after the bar code scanner has been delivered.)
Status	The progress of the user story such as “Defined”, “To Do”, “In Progress”, “Tested”, “Done”, “Removed”.
Creation Date	The date of user story creation.
Last Updated Date	The last updated date of user story.

7.5 S5 - STORY BOARD

Based on the project progress updated at the PRL in Figure 6, a story board will be generated automatically as shown in Figure 8 below.

[Project Name] - Story Board					
Defined	To Do	In Progress	Testing	Done	Removed
			Computer Items Lending		
		Login			
	Reset Password				

Figure 8 - A Story Board

Table 13 - A Story Board Description

Column Name	Description
Defined	Outstanding user story.
To Do	The task was assigned to a programmer/SA and planned to be done at the current timebox.
In Progress	The task was being implemented.
Testing	Testing was being performed.
Done	The user story was completed and tested.
Removed	The user story was removed (i.e. withdrawn) from the PRL.

7.6 S6 - TASK BOARD

Based on the project progress updated on the detailed tasks list in Figure 7, a task board will be generated automatically as shown in Figure 9 below.

[Project Name] - Task Board - Timebox 3				
Corresponding Story	To Do	In Progress	Testing	Done
Computer Items Lending				Layout design
			Backend logic on lending	
		Validation rules creation		
Login	Form design and creation			
	SQL creation			
Reset Password	Layout design			
	SQL enquiry			
	Backend logic on reset and return password			

Figure 9 - A Task Board

Table 14 - A Task Board Description

Column Name	Description
Corresponding Story	The user story for the detailed task.
To Do	The detailed task was assigned to a programmer/SA, and planned to be done at the current timebox.
In Progress	The task was being implemented.
Testing	Testing was being performed.
Done	The user story was tested and completed.

8 DESIGN THINKING

Design Thinking (DT) is an approach of understanding customer problems and innovating solutions for them. DT empathises with customer needs, generates manifold possible solutions, creates multiple prototypes, and tolerates failures to ultimately arrive at an end products or services that offer customers simple solutions to their unmet or real needs.

For DT, the members of cross functional teams perform user research, gather business needs and discuss technology possibilities. For Agile, this translates into creating backlogs, writing user stories and determining success metrics. Both embraces iteration as part of the process and therefore establish ongoing refinement to optimise the business value. Both are stimulated by members in cross-functional teams which stimulate cross-functional solutions for products and services.

Table 15 - Five Steps in Design Thinking

Empathise & Observe	<p>What does customer care about most?</p> <p>Immerse in the experience of the user and understand through observation and interaction. Identify the emotional as well as the functional needs of the users— what are their desires, aspirations, concerns?</p>
Define	<p>What are customer's unmet needs</p> <p>Define a problem to tackle. Seek to frame efforts so as not to limit ideas and define the problem and arrive at a problem statement which will give the necessary direction to proceed towards the issue faced by the customer.</p>
Ideate	<p>What product or service would most effectively meet customer's unmet or real needs?</p> <p>Diversify and explore a large variety of ideas. Encourages unconstrained ideation that is, the encouragement of all ideas no matter how unconventional. DT attempts to avoid that by opening up the ideation stage across business functions and inviting everyone to suggest solutions.</p>
Prototype	<p>How can I best illustrate my solution to my customer?</p>

	<p>Create the ideas to tangible forms so that users can experience and interact with it. Prototype the best and most promising concepts. In this stage, instead of a sophisticated prototype, designers should be empowered to create any number of rough prototypes. This rapid, iterative process filters out unsuccessful ideas quickly, leaving room for more promising ideas to take shape.</p>
Test & Refine	<p>What worked? What didn't work? How can I make it better?</p> <p>The idea that seems the best according to the feedback of the customers and end users in the prototype phase will be executed. In this step, the design thinkers are supposed to be collaborative and agile. After testing, the entire process of DT may have to be repeated. If the end user approves the solution, then the process of DT stops here.</p>