

## NashTech Software Development

### Scrum Process NashTech



Version 2.1.1

August 10<sup>th</sup> , 2016

110 Bishopsgate  
London  
EC2N 4AY

Tel: +44 (0)20 7333 0033  
Fax: +44 (0)20 7333 0032

Security Classification: **Internal**

## **TABLE OF CONTENTS**

+ Add Product Integration and Release Procedures .....	3
1 INTRODUCTION .....	1
1.1 Purpose .....	1
1.2 Scope .....	1
1.3 Definitions, Acronyms and Abbreviations .....	1
1.4 References .....	1
2 Scrum Theory .....	2
2.1 Transparency.....	2
2.2 Inspection .....	2
2.3 Adaptation .....	2
3 Scrum Methodology .....	2
3.1 Scrum Phases .....	3
4 NashTech Scrum Overview .....	5
4.1 Criteria for NashTech Scrum Approach .....	6
5 NashTech Scrum Process .....	7
5.1 Scrum Rule:.....	7
5.2 Scrum Planning: Sprint 0 .....	7
5.3 Development (Sprint N + 1 (N = 0, 1, 2, 3....)) .....	9
5.4 Cancelling a Sprint.....	12
5.5 Closure .....	12
5.6 Definition of Done .....	12
6 Product Integration and Release Procedures.....	12
6.1 Integration.....	12
6.2 Release .....	12
7 Steps to estimate a ROM for a user story .....	13
7.2 Scrum Artifacts .....	14
8 NashTech Scrum Roles – Scrum Team .....	14

## RECORD OF CHANGE

Document Location	
File Name	Location
HNVN_SD_Scrum_HarveyNash_Scrum_Process	<a href="https://intranet.harveynash.vn/helpdeskblog/hvnPAL/Pages/Forms/AllItems.aspx?RootFolder=%2Fhelpdeskblog%2FhvnPAL%2FPages%2FProcess%20Access%20Library%2FSCRUM%2FProcess">https://intranet.harveynash.vn/helpdeskblog/hvnPAL/Pages/Forms/AllItems.aspx?RootFolder=%2Fhelpdeskblog%2FhvnPAL%2FPages%2FProcess%20Access%20Library%2FSCRUM%2FProcess</a>

Document Version History			
Version	Date	Author	Details
0.1	18 October, 2011	Ly Vo, QAM	Initialized the document
1.0	23 November, 2011	Ly Vo, QAM	Updated 4.1
1.1	23 April, 2013	Thao Phan	Update 5.2.1 Planning: Get SOW from EM and Conduct Green Hat meeting using Green Hat checklist
1.1.1	20 February, 2014	Huyen Dinh	Update 4: link of tailoring process
1.2	29 June, 2014	Tham Tran	Update format, style guide
1.3	15 Oct, 2014	Huyen Dinh	Add Daily Report in output list of Daily meeting
2.0	10 Dec,2014	Huyen Dinh	Update 4.1: Criteria for NT Scrum Approach Make clear Scrum meetings;
2.1	16 Jun,2015	Hien Nguyen, Minh Bui	Update + Update Scrum rule to mention SLA and RADIO into Scrum process + Use SAD to replace Design Note document + Add Coding Process in Sprint 0 (5.2) + Add Product Integration and Release Procedures
2.1.1	10 Aug,2016	Trang Le	Change “Genkiens” to “Jenkins”

### APPROVERS

Distribution List				
Name	Role	Signature	Date	Version
Cuong Nguyen	Managing Director	Signed	23 November 2011	1.0
Ly Vo	QAM	Signed	15 December 2014	2.0

### CONFIDENTIALITY

This document is distributed on a restricted basis, is commercial in confidence to the recipient, and may not be used for any purpose other than that associated with a NashTech project. The contents of this document may not be disclosed to any third parties without the expressed advance written authorisation of NashTech.

## 1 INTRODUCTION

### 1.1 Purpose

The purpose of this document is to help projects to compliant with NashTech Standard Process when apply Scrum Approach in NashTech.

This document is a part of Software Process Improvement in NashTech Standard Process

### 1.2 Scope

This process is to describe when should use Scrum Approach in NashTech and what should apply if approaching Scrum in NashTech at the beginning to the end of project.

### 1.3 Definitions, Acronyms and Abbreviations

Term	Explanation
CM	Configuration Management
CMO	Configuration Management Officer
ODC	Offshore Development Centre
T&M	Time and Material
Q&A	Questions and Answers
QA	Quality Assurance
QAM	Quality Assurance Manager
QAO	Quality Assurance Officer
QC	Quality Controller
NC	Non-compliance
ROM	Rough Order of Magnitude
SAD	Software Architecture Design

### 1.4 References

No	Document Name	Ref to:
1	<a href="http://planningpoker.com">http://planningpoker.com</a>	<a href="http://planningpoker.com">http://planningpoker.com</a>
2	Scrum Primer Version 1.2	Pete Deemer - Scrum Training Institute (ScrumTI.com) Gabrielle Benefield -Scrum Training Institute (ScrumTI.com) Craig Larman -craiglarman.com Bas Vodde - Odd-e.com
3	Scrum Guide 2011	Ken Schwaber and Jeff Sutherland

No	Document Name	Ref to:
4	SCRUM Development Process	Ken Schwaber

## 2 Scrum Theory

*Notes: To Extracted from Scrum guide -2011 of Ken Schwaber and Jeff Sutherland*

Scrum is founded on empirical process control theory, or empiricism. Empiricism asserts that knowledge comes from experience and making decisions based on what is known. Scrum employs an iterative, incremental approach to optimize predictability and control risk. Three pillars uphold every implementation of empirical process control: **transparency, inspection, and adaptation.**

### 2.1 Transparency

Significant aspects of the process must be visible to those responsible for the outcome. Transparency requires those aspects be defined by a common standard so observers share a common understanding of what is being seen. For example:

- A common language referring to the process must be shared by all participants via kick off meeting or sprint planning; and,
- A common definition of “Done” (refer to 5.6) must be shared by those performing the work and those accepting the work product.

### 2.2 Inspection

Scrum users must frequently inspect Scrum artifacts and progress toward a goal to detect undesirable variances. Their inspection should not be so frequent that inspection gets in the way of the work. Inspections are most beneficial when diligently performed by skilled inspectors at the point of work.

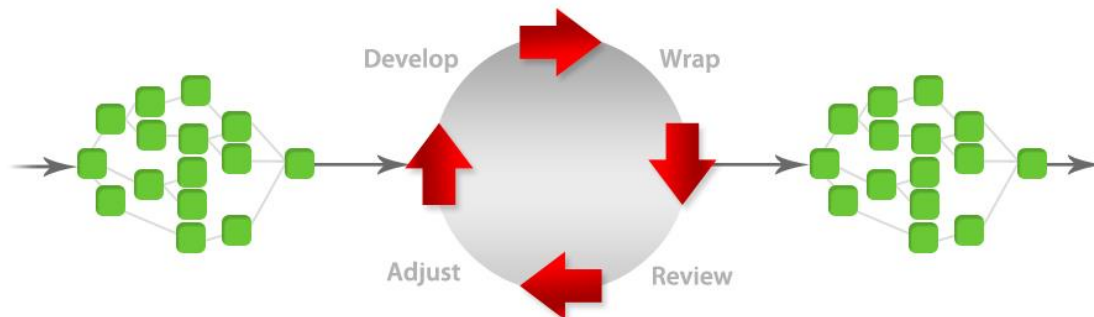
### 2.3 Adaptation

If an inspector determines that one or more aspects of a process deviate outside acceptable limits, and that the resulting product will be unacceptable, the process or the material being processed must be adjusted. An adjustment must be made as soon as possible to minimize further deviation

## 3 Scrum Methodology

*Note: To be extracted from SCRUM Development Process Ken Schwaber*

## Harvey Nash Scrum Methodology



**Sprint 0 Planning  
& System Architecture**

**Sprints**

**Closure**

### Characteristics of SCRUM methodology are:

- The first and last phases (Planning and Closure) consist of defined processes, where all processes, inputs and outputs are well defined. The knowledge of how to do these processes is explicit. The flow is linear, with some iteration in the planning phase.
- The Sprint phase is an empirical process. Many of the processes in the sprint phase are unidentified or uncontrolled. It is treated as a black box that requires external controls. Accordingly, controls, including risk management, are put on the iteration of the Sprint phase to avoid chaos while maximizing flexibility.
- Sprints are nonlinear and flexible. Where available, explicit process knowledge is used; otherwise tacit knowledge and trial and error is used to build process knowledge. Sprints are used to evolve the final product.
- The project is open to the environment until the Closure phase. The deliverable can be changed at any time during the Planning and Sprint phases of the project. The project remains open to environmental complexity, including competitive, time, quality, and financial pressures, throughout these phases
- The deliverable is determined during the project based on the environment

### 3.1 Scrum Phases

#### 3.1.1. Pregame (Sprint 0)

##### Planning:

Definition of a new release based on currently known backlog, along with an estimate of its schedule and cost. If a new system is being developed, this phase consists of both conceptualization and analysis. If an existing system is being enhanced, this phase consists of limited analysis.

##### Architecture:

Design how the backlog items will be implemented. This phase includes system architecture modification and high level design.

### 3.1.2. **Game**

#### **Development Sprints:**

Development of new release functionality, with constant respect to the variables of time, requirements, quality, cost, and competition

Interaction with these variables defines the end of this phase. There are multiple, iterative development sprints, or cycles, that are used to evolve the system.

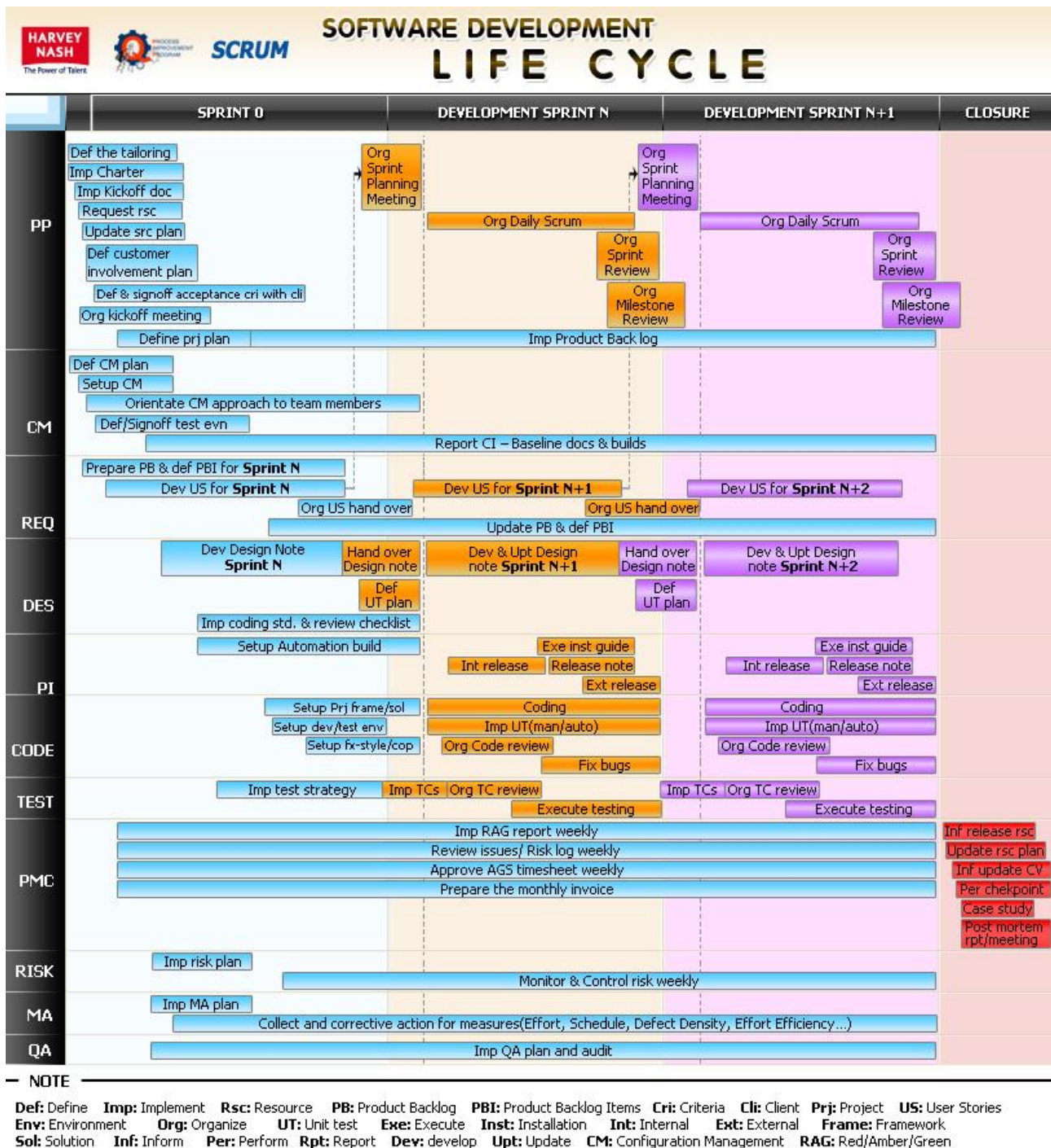
### 3.1.3. **Postgame**

**Closure:** Preparation for release, including final documentation, pre-release staged testing, and release.



#### 4 NashTech Scrum Overview

The NashTech Scrum approach is embraced both Agile methodology and CMMI model to make sure agile deliverables in 2-8 weeks can be quality implemented and adds the best possible value to organization and customer.



## 4.1 Criteria for NashTech Scrum Approach

Scrum approach is suitable for projects have following characteristics.

1. Project type is not fixed price. It should be ODC, T&M
2. Client and Project team members (Scrum team) have good awareness on SCRUM.
3. Scrum Team: Product Owner, Scrum master, Development Team are clearly defined.

4. Product Owner is **representative of client or client** and make sure they have time to stay with development team for queries and explanation on stories. (It's should be at least 2 hours/day)
5. Development team are self-organising

## 5 NashTech Scrum Process

### 5.1 Scrum Rule:

- Scrum team must be aware of project scrum methodology
- The Sprints are time-boxed, they end on a specific date whether the work has been completed or not, and **are never extended**.
- SAD must be ready at sprint 0. SAD is not required to provide full views / diagrams but good enough to describe the current system or the system to be implemented.
- SAD must be updated frequently once there is any change impacts to the current architecture.
- Design note must be provided for user stories base on the criteria defined by project and approved by technical team.
- Technical Architect in responsible for revise anything impacts to the system.
- Technical Architect or Team Lead is in charge of updating documents
- If project does not have Technical Architect or Team Lead, Project Manager will specify a person to be responsible for updating documents.
- Sprint scope must be finalized and sent out after sprint planning
- Channel to explain user story is clearly defined. Ex: Document, tool,
- Determine whether or not any remaining or new work should go into the backlog for the next iteration
- Daily stand-up meeting must be starts on **time around 15 minutes**. All are welcome, but normally only Product Owner, Development Team, Scrum Master speaks, and other should contribute ideas/suggestion after the meeting with scrum master, project manager. The meeting should happen at the same time every day
- The Product Owner is one person.
- All work products of project must be produced following Nash Tech Standard Service Level Agreement (SLA) for Scrum.
- All work products of project must be reviewed and approved by appropriate roles, following the definition in Nash Tech Standard Software Development RADIO for Scrum.
- Processes (such as Project Monitoring and Control, Coding, Testing process...) which are applied in Scrum project are referred to standard processes. And it may be tailored to match with project's characteristic. (*Refer to tailoring sheet for SCRUM*)

### 5.2 Scrum Planning: Sprint 0

**Purpose:** To plan and design Architecture high level to prepare getting start for next sprints.

#### 5.2.1. Planning

- Duration is about 6-8 weeks.  
**Project manager** is to prepare following activities to getting start for next sprints
  - o Get SOW from EM
  - o Conduct Green Hat meeting using Green Hat checklist
  - o Develop Project charter (resource, stakeholder, project approach, communication plan, role & responsibility.)
  - o Define Tailoring process (*Refer to tailoring sheet for SCRUM*)
  - o Define Risk plan and Risk log

- Define all sub-plan (Measure Objective plan, Configuration Management Plan, Test strategy, Quality Assurance Plan)
- Define Definition of Done for User story and Successful Sprint
- Define Product Owner Commitment Conduct Kick-off Meeting to transfer the scope, vision, role & responsible, process approach, quality objectives, lesson learnt,...to project team.

### **Configuration Management Official (CMO)**

- Produce CM plan
- Prepare development, test, deployment, CI frameworks and environments

### **Scrum Master** is to prepare

- Coaching project team to understand Scrum Methodology.
- Verify and make sure the activities of planning are done.
- Verification of management approval and funding.

## **5.2.2. Requirement, Design & Coding**

### **Product Owner** is to

- Gather the business requirements, produce the product backlog and be responsible for User stories approval.
- Enough of the product backlog must be estimated (refer to 6 below) and prioritized that the team can pick a reasonable amount of high value work from the backlog **in the first Sprint Planning**
- High level vision of expected outcome and implementation (scope of project)
- Define “Definition of Done” for each user stories refer to sample in 5.6
- Estimation (refer to 6) of release cost, including development, collateral material, marketing, training, and rollout.

### **Development team to prepare following activities:**

Architecture vision should be initialized and discussed. First version of high level architecture should be discussed and documented down (the document does not need contain all the following items but it must be good enough to describe the overview system)

- Key technical challenges, constraints
- High level system overview
- High level architectural break down with requirement and solution
- High level components view
- Touch points, integration points between subsystems or sub components.

Design and coding standards, coding review checklist need to be tailored and approved. They should includes

- Customized rules set for automatic design and coding metrics analysing tools
- Design and code guidelines and review checklist for each type of component or technologies used. For example: design guideline and checklist for database, design and implementation guideline and check list for security ...

Base framework and solution structure and related documents also need to be ready.

Optionally draft plan for key design activities need to be defined and tracked as technical spikes

Unit test plan is not required to get approval in Sprint 0. Team can use document to track unit test plan or create sub task in each User Story.

*Note: (Refer to HNVN\_SD\_012\_Guideline\_Design\_For\_Scrum.docx for further information)*

**Configuration Management Official (CMO)** is to baseline the package (all plans, user stories, design notes).

### 5.2.3. Product Integration

Development team (PM to specify one) to prepare Product Integration (PI) environment by following Product Integration procedures mentioned below to make sure the PI will be done in every sprint.

During the sprint 0, the development team needs to develop the build script so as to have the continuous integration system setting up.

The build script should include below activities:

- + Compile code
- + Run Unit Test
- + Check static tool
- + Installing package
- + Auto deployment if needed
- + Run automated system test (functional, and non-functional(load test, performance)) if any.

The build guide needs to be documented after the build script is finalized.

Build plan document is not required to use in Scrum process

### 5.3 Development (Sprint N + 1 (N = 0, 1, 2, 3....))

- Iterative with short life cycles (2→4 weeks sprint and schedule of software release) enabling fast verifications and corrections.
- Product is delivered frequently for checkpoint / milestone verify as well as close monitor control of any issue arise.
- The Sprints are time-boxed, they end on a specific date whether the work has been completed or not, and are **never extended**.
- Sprints contain and consist of the Sprint Planning Meeting, Daily Scrums, the development work, the Sprint Review Meeting, and the Sprint Milestone
- Sprint planning is about 4-8 hours.
- Daily stand-up meeting must be starts on **time around 15 minutes**
- Sprint Demo is about **4hours for one – month sprints**.
- Retrospective Meeting is about **3hours time-boxed for one - month sprints**
- Requirement, design notes need to create, update for next sprint.

#### 5.3.1. Sprint Planning Meeting

**When:** At the beginning of each Sprint

**Purpose:** the Development Team works to forecast the functionality that will be developed during the Sprint. Product owner presents ordered Product Backlog items and for how to implement the items that the Team decides to take on.

**Input:** Make sure all input followings are ready for sprint planning in the part one.

- **The product backlog**
- **Q&A, User stories, design note**
- The latest product increment
- Project capacity of Development Team
- Past performance of Development Team (Refer to burn down chart and Productivity/Velocity)

**Discussion in meeting**

- **Clarify the selected Product Backlog items:** Discuss the goals and context for these high-priority items on the Product Backlog, providing the Development Team with insight into the Product Owner thinking.
- **Breakdown US into tasks.** Task for a US should be included review code, unit test, fix bug task
- The Product Owner and Development Team also review the “Definition of Done” refer to 5.6 below
- Plan to audit the sprint demo by QAO
- Plan to study user stories for next sprint.
- The Development Team should be able to explain to the Product Owner and Scrum Master how product backlog item intends to work as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment
- To make sure all items in checklist ready is discussed.

### Output:

- Items will be delivered in the sprint
- The Development team understands the goal.
- Product backlog has been updated
- Updated DoD for US and Sprint if needed.
- Sprint’s scope/goal
- Updated Sprint log

### 5.3.2. Daily Meeting

**When:** Every day in Sprint

**Purpose:** Scrum Master to review the Sprint Goal and to synchronize activities and create a plan for the next 24 hours

### Input:

- The Development Team selects the items from the Product Backlog

### Discussion in the meeting

- Each Development Team member explains:
  1. What has been accomplished since the last meeting?
  2. What will be done before the next meeting?
  3. What obstacles are in the way?
- Other can be joined daily meeting as chicken role (listen, not speaking). They can discuss with related person to raise issue, risk, suggestion ... after a meeting.

### Output:

- Improve communications,
- Eliminate other meetings,
- Identify and remove impediments to development, highlight and promote quick decision-making,
- Improve the Development Team’s level of project knowledge.
- Updated issue and risk log Sprint Daily Report (optional)



### 5.3.3. Sprint Demo Meeting

**When:** At the end of Sprint

**Purpose:** to inspect the Increment and adapt the Product Backlog if needed and to elicit feedback and foster collaboration.

**Input:**

- QC report
- QA Audit report (e evaluate what US meets DoD to demo and evaluate Sprint is successful or pass based on DoD of Sprint )
- Baselined the package
- Release Note

**Discussion in the meeting**

- The Product Owner identifies what has been “Done” and what has not been “Done”;
- The Development Team demonstrates the work that it has “Done” and answers questions about the Increment;
- The entire group collaborates on what to do next, so that the Sprint Done provides valuable input to subsequent Sprint Planning Meetings.

**Output:**

- Revised Product Backlog that defines the probable Product Backlog items for the next Sprint.
- Updated Release note
- Updated issue and risk log

### 5.3.4. Sprint Retrospective Meeting

**When:** After the Sprint Demo meeting and prior to the next Sprint Planning Meeting

**Purpose:**

- Inspect how the Sprint went with regards to people, relationships, process, and tools
- Identify and order the major items that went well and potential improvements;
- Create a plan for implementing improvements to the way the Scrum Team does its work.

**Input:**

- 
- Retrospective report
- Measurement data of Sprint.
- Lesson learnt, best practise

**Discussion in the meeting**

- The Scrum Team plans ways to increase product quality by adapting the Definition of Done” as appropriate.
- Development Team member focus on:
  - + What went well during the last sprint?
  - + What could be improved in the next sprint?

+ Action need

### Output:

- Retrospective meeting report

### 5.4 Cancelling a Sprint

If the Sprint Goal becomes obsolete such as changes direction or market or technology conditions change. It's no longer make sense given the circumstances, a Sprint can be cancelled. Only Product Owner has authority to cancel the Sprint.

#### In this case, project team will:

- Review the work items in Sprint.
- Re-estimated incomplete work items
- Re set priority and put back on the product backlog
- Everyone has to regroup in another Sprint
- Release resource if needed

### 5.5 Closure

*Refer to Closure procedure in HNVN\_SD\_014\_Process\_ProjectMonitoringAndControl.docx*

### 5.6 Definition of Done

Definition of Done is a comprehensive checklist of necessary activities to ensure that only truly done features are delivered, not only in terms of functionality but in terms of quality as well (DoD of for a Product Backlog Item/User Story). Beside that it define the criteria to evaluate a Sprint is successful or fail (DoD of Sprint)

Refer HNVN\_SD\_026\_Scrum\_Template\_DefinitionOfDone

## 6 Product Integration and Release Procedures

### 6.1 Integration

The source code is expected to be built after there is any change happened. This build could be configured automatically using tool like Jenkins to happen rightly, daily, twice a day or being done manually.

When the build is failed; the warning (like email) should be sent to the Leader and PM.

Newly build is required to perform smoke test before release to QC. Which case has been tested and smoke test result (pass and fail) will be sent to project team via email, predefined template... Bug of smoke test is not required to log.

### 6.2 Release

When it comes to release a specific build to the test team the below steps should be applied:

- CMO takes the build and install it to the testing environment
- CMO conducts the smoke testing (using predefined smoke test cases)
- If smoke test cases are passed. CMO issues the release notes to the test team by email



- If failed CMO reports to PM and reject the build.

The above steps could be done automatically using tools (like Jenkins)

### 7 Steps to estimate a ROM for a user story

SCRUM projects can be estimated using standard function point estimating. However, it is advisable to estimate productivity at approximately twice the current metric. Steps are following.

1. Implement some first sprints (suggest 2 or 3) with some user stories.
2. These user stories in these sprints must not too small or large (suggest 1-4 working days of development for each)
3. The Product Owner (PO) works with scrum members to choose based user story
4. The based user story must have below properties:
  - Its size is not too large or small (suggest around 2-3 working days for 2-3 developers)
  - It should include a very common feature of the product (like add, edit, delete, view, report...)
  - All developers should provide the effort for each task break down very easily and shortly.
  - Development team must provide a precise estimate (effort) on how much does it take for development and testing that user story.
5. The PO marks the ROM (Rough Order of Magnitude) of that basic user story is 3.
6. Before estimating the ROM value for the new user stories, the PO send all available user stories to scrum member to read
7. After reviewing, each scrum member makes him/herself ROM estimate base on the comparison to the based user story (on step above)
8. The PO organizes the ROM meeting for new user stories (in step #6 above) using planning poker tool at <http://planningpoker.com/>
9. After the meeting session on step #8 above, each new user store will have its own ROM value (compare with value 3 of based user story)
10. Note: the PO need to break down the user story if the team agree its ROM value is greater than 8 (means that user story is too big for a sprint)

#### 7.1.1. Breaking user story to smaller user stories

If a user story does not meet the below standards, the story may be too large, the team should break it into smaller stories.

- Independent. User Stories should be as independent as possible.
- Negotiable. User Stories are not a contract. They are not detailed specifications. They are reminders of features for the team to discuss and collaborate to clarify the details near the time of development.
- Valuable. User Stories should be valuable to the user (or owner) of the solution. They should be written in user language. They should be features, not tasks.
- Estimable. User Stories need to be possible to estimate. They need to provide enough information to estimate, without being too detailed.
- Small. User Stories should be small. Not too small. But not too big!
- Testable. User Stories need to be worded in a way that is testable, i.e. not too subjective and to provide clear details of how the User Story will be tested.
- In each sprint, there should be around 4 to 6 stories included, if the number of stories in a sprint is small, if your team could not finish a story, it will highly affect to your productivity.

### 7.1.2. Breaking user story to tasks

- Include all tasks necessary to make the Product Backlog item **100% complete** – i.e. potentially shippable – within the Sprint
- Tasks may include the traditional steps in a development lifecycle (although limited to the feature in question, not the entire product). For instance: Design, Development, Unit Testing, System Testing, UAT (User Acceptance Testing), Documentation, etc.
- Working time of each task should be from 2 hours to 2 days as small task will be easy to estimate and control.

## 7.2 Scrum Artifacts

Refer to HNVN\_SD\_016\_07\_Scrum\_TailoringProjectProcess.xlsx

## 8 NashTech Scrum Roles – Scrum Team

### 8.1.1. The Scrum Master

- Coaching the Project Team in self-organization and cross-functionality;
- Teaching and leading the Project Team to create high-value products;
- Removing impediments to the Project Team's progress
- Helping the project team and stakeholders understand and enact process and empirical product development;
- Causing changes that increase the productivity of the Project Team and the effectiveness of the product.

### 8.1.2. Project Manager

- Doing tailoring for project.
- Overall project delivery, building working environment, resource management and also facilitating the communication.
- Planning and implementing the product so that it fits within our organization's culture and still delivers the expected benefits
- Ensuring that the Project Team adheres to project approach, process activities, rules – policy, roles and responsibilities in project.
- Performing project management activities
  - Ensuring that requirement documents are based-line and arranged for requirement review meetings.
  - Review Acceptance Criteria for each sprint and each release
  - Ensuring Monitoring and Controlling activities runs as defined
  - Ensuring the project is controlled in good health which means values of parameters should meet the defined objectives (effort, schedule, defect...). Otherwise, it must be followed up for adjustments.
  - Ensuring CM activities and CR procedure
  - Performing the risk and issue management
    - Identifying and document risks within the project.
    - Submitting risks to EM, Delivery Manager and Product Owner for review.
    - Monitoring the progress of all risk mitigating actions assigned
  - Ensuring each delivery is released on time (no delay) with the highest quality
  - Defining Project Measurement Objectives (defect density, defect leakage, velocity, effort), Plan to monitor and control the measurement objectives.
  - Collecting Lesson Learnt & Best Practice and conducting Milestone Report

- Conduct technical meetings with Technical team to discuss about technical aspects in term of architecture and technical challenges.

To ensure that everyone follows Process activities and Project rules.

### 8.1.3. The Product Owner

The Product Owner, who is representative of client or normally is business analysis, has responsible for

- Representing the interests of all the stakeholders
- Ensuring business requirements & objectives are met
- Reviewing and signing off product specification requirements, project deliverables, the final technical solution and acceptance criteria
- Being available to answer project teams' questions and be ready to give advice.
- Clearly expressing Product Backlog items;
- Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Development Team will work on next
- Ensuring the Development Team understands items in the Product Backlog to the level needed.
- Managing Product Backlog
- Leading the development effort towards the project's success and maximize projects' effectiveness by
  - o Creating the project's initial overall requirements, high-level feature requirements, etc...
  - o Ordering the items in the Product Backlog to best achieve goals and missions
  - o Defining the release plans after having rough relative size estimates
  - o The Product Owner is one person, not a committee. The Product Owner may represent the desires of a committee in the Product Backlog, but those wanting to change a backlog item's priority must convince the Product Owner

### 8.1.4. The Development Team

- Perform the coding and follow project development process to deliver potentially releasable Increment of "Done" product at the end of each Sprint. Only members of the Development Team create the Increment.
- Carries out unit testing
- Self-managing, self-organizing: No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality.
- Cross-functional with all of the skills as a team necessary to create a product Increment
  - o Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole; and,
  - o Development Teams do not contain sub-teams dedicated to particular domains like testing or business analysis.
  - o Scrum recognizes no titles for Development Team members other than Developer, regardless of the work being performed by the person;

*Note: Although the best thing is to have cross-functional team, development team can be structured like below*

- *Developers: Perform the design, coding, unit test, review code.*
- *Testers: Plan to test, produce test case, execute test and test report,*

### 8.1.5. The Business Analysis

- Assist Product Owner in all initiatives related to define user stories.
- Assist the development team understand the requirement of project
- Join in all meeting of projects.
- Raise business concerns.

### 8.1.6. Quality Assurance

- Support project team members to understand process, template, guideline using in the project, and;
- Make sure offshore/onshore team follow strictly processes agreed in tailoring sheet of the project.
- Audit project before sprint demo and audit other area which includes Project planning, monitoring & controlling, risk, issue, configuration management activities, measurement activities 3-weeks recurring.

### 8.1.7. CMO

- Produce CM plan
- Prepare your development, test, deployment, CI frameworks and environments
- Perform baseline before sprint demo
- Release note

### 8.1.8. Client PM

- Plans, initiates and closes projects
- Deals with risks, issues and dispute resolution
- Tracks progress against plan (although some day to day operational matters are delegated to the offshore team leaders)
- Receive the progress report from offshore team.

### 8.1.9. Engagement Manager

- Focus on Operations
- Works with client team to plan for the medium and long term
- Escalation point for Offshore PM
- Escalation point for client PM
- To be chair the regular Checkpoint meeting between client and offshore delivery team.

### 8.1.10. Technical Team

- In charge of technical owner who will sign off technical things of offshore delivery.
- Support Development team to defines the architectural roadmap and the development tools and environment
- Provides technical support and technical quality control throughout projects