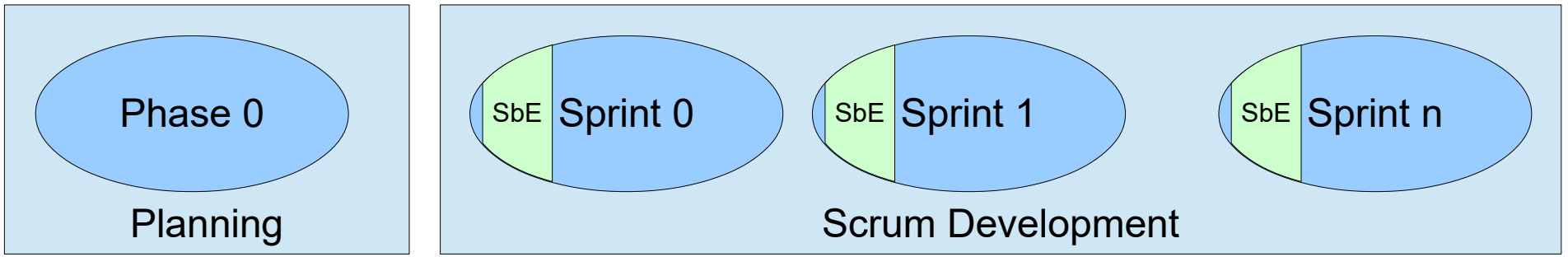
Requirements Team

Phase 0
Specification by Example

Overview

- Phase 0: Planning phase before development starts
- Sprint 0: Initial technical development sprint to set up technical infrastructure and system architecture, developing only one or a few user stories
- Specification by Example: Each sprint starts with defining and automating concrete examples of how the system should behave

Timeline


Phase 0
Planning
SbE
Sprint 0
SbE
Sprint 1
SbE
Sprint n
Scrum Development

Planning?

- Big Design Up Front = BAD!
- YAGNI

Planning?

- However, having an idea of what you are going to build, and how, is useful.
- Big Picture Up Front = GOOD!

Failure Factors in Agile

- Chow, T., & Cao, D. (2008). A Survey Study of Critical Success Factors in Agile Software Projects.
- Identifies 18 failure factors in agile development projects, divided into four dimensions:
  - Organisational
  - People
  - Process
  - Technical

Failure Factors in Agile

- Organisational
  - Lack of executive sponsorship
  - Lack of management commitment
  - Organisational culture being too traditional
  - Organisational culture being too political
  - Organisation being too large
  - Lack of agile logistical arrangements
- People
  - Lack of the necessary skill set
  - Lack of project management competence
  - Lack of team work
  - Resistance from groups or individuals
  - Bad customer relationship
- Process
  - Ill-defined project scope and project requirements
  - Ill-defined project planning
  - Ill-defined customer role
  - Lack of agile progress tracking mechanisms
  - Lack of customer presence
- Technical
  - Lack of correct agile practices
  - Inappropriate technology and tools

Failure Factors That Phase 0 Helps You With

- Organisational
  - Lack of executive sponsorship
  - Lack of management commitment
  - Organisational culture being too traditional
  - Organisational culture being too political
  - Organisation being too large
  - Lack of agile logistical arrangements
- People
  - Lack of the necessary skill set
  - Lack of project management competence
  - Lack of team work
  - Resistance from groups or individuals
  - Bad customer relationship
- Process
  - Ill-defined project scope and project requirements
  - Ill-defined project planning
  - Ill-defined customer role
  - Lack of agile progress tracking mechanisms
  - Lack of customer presence
- Technical
  - Lack of correct agile practices
  - Inappropriate technology and tools

Failure Factors that SbE Helps You With

- Organisational
  - Lack of executive sponsorship
  - Lack of management commitment
  - Organisational culture being too traditional
  - Organisational culture being too political
  - Organisation being too large
  - Lack of agile logistical arrangements
- People
  - Lack of the necessary skill set
  - Lack of project management competence
  - Lack of team work
  - Resistance from groups or individuals
  - Bad customer relationship
- Process
  - Ill-defined project scope and project requirements
  - Ill-defined project planning
  - Ill-defined customer role
  - Lack of agile progress tracking mechanisms
  - Lack of customer presence
- Technical
  - Lack of correct agile practices
  - Inappropriate technology and tools

Scrum Phases and Processes

- According to SCRUMstudy:
- Scrum processes address the specific activities and flow of a Scrum project. In total there are 19 processes in SBOK™ Guide which are grouped into following five phases:
  - Initiate
  - Plan and Estimate
  - Implement
  - Review and Retrospect
  - Release

Processes in the Initiate Phase

- Create Project Vision
- Identify Scrum Master and Stakeholder(s)
- Form Scrum Team
- Develop Epic(s)
- Create Prioritized Product Backlog
- Conduct Release Planning

Phases + Processes = Agile?

- You may question if software development with well-defined phases and processes is really agile.
- If you always follow the prescribed set of activities, this is not agile.
- However, a checklist to help you remember important activities is useful.
- Remember to always question whether a specific artifact is required in your case.
- Also remember that an artifact is never frozen, it can always be updated if necessary. This is true for everything, including architecture, for example.
- The artifacts you produce should be as light-weight as possible.

Phase 0 Artifacts

- The following is an uncomplete list of artifacts that are useful to consider in phase 0:
  - Vision and goals
  - List of stakeholders and all Scrum roles (Scrum team, Scrum master, product owner)
  - Initial architecture
  - Roles/personas
  - Initial product backlog
  - (Release plan)
  - (Paper prototype of user interface and other usability artifacts)

Vision and Goals

- A product vision acts as the project's true north, sets the direction and guides the Scrum team. It is the overarching goal everyone must share.
- The product vision paints a picture of the future that draws people in.
- It describes who the customers are, what customers need, and how these needs will be met.
- It captures the essence of the product – the critical information we must know to develop and launch a winning product.
- "Can you explain your product in the time it takes to ride up in an elevator?"

Vision and Goals

- Goals should be SMART:
  - Specific (clear, concise, tangible)
  - Measurable (dollars, volume, time, experiences)
  - Attainable (you can do something to actually make this happen)
  - Realistic (are you willing and able to do this?)
  - Timely (deadlines are announced and committed to)
- Impact mapping can help define the goals for a project.

Stakeholders

- It is extremely important to find everyone affected by the new system as soon as possible, in order to get their input.
- Everybody, all together, from early on! is the secret of Lean, according to the book Lean Architecture

Stakeholders: Who to Include?

- The product owner and the team members, such as developers and testers, are obvious.
- Architects will want a say in how the system is designed.
- System administrators want to know how the system affects IT operations.
- The support organization may need to learn a bit about the system to be able to answer questions.
- If the system requires training, the people developing the training material need to be informed as soon as possible.
- The sales department may need to know about the system or product being built.
- Last but not least, the users of the system may be interested in what the future brings.

Initial Architecture

- It is good if you have some idea of the high-level design of the system before you start developing it.
- For example: which databases and message queues are involved; which external or internal systems do you need to integrate with; what are the most important services in your system?
- This can be a drawing on a whiteboard (or a napkin). It may be useful to put a copy in a wiki for easy access.

Roles/Personas

- A system has many different types of users, or roles, that use the system differently.
- For a job posting and search site, some possible roles are:
  - Job seeker
  - First timer
  - Layoff victim
  - Job poster
  - Resume reader

Roles/Personas

- For some systems, especially systems where user experience is very important, you may want to make up pretend users and design for them.
- These pretend users are called personas.
- Each persona is named and described with a fictitious biography. They may even have a picture.

Roles/Personas

- It is very important that every user story states which role or persona uses the system, why they do it, and what benefits they gain from this:
  - As a <role/persona> I want to <goal> so that <benefit>
- Mike Cohn even suggests deriving the user stories from the goals of the personas.

Initial Product Backlog

- Before you can start developing user stories in a development sprint, you need to have an idea of which the most important user stories are.
- The user stories and epics (high-level user stories that probably must be broken into smaller stories) are documented in a product backlog, using post-it notes, or a software system such as JIRA or Trello.

Release Plan

- It may be necessary to estimate how long development will take, or how many user stories will be finished in a given time.
- This requires an estimated product backlog, and an estimate of the velocity of the team.
- To create these estimates early in the project is extremely difficult.
- If you don't have to estimate, then don't.
- Sometimes you can simply work for a number of sprints and hopefully develop something useful, and then decide if development should continue or not.
- Estimates made later in the project are much more precise than the ones made before development starts.

Paper Prototype of UI and Usability Artifacts

- If the system you are building has a complex user interface, it may be useful to early create a lofi prototype of the interface, using a whiteboard or paper.
- If you are serious about usability, phase 0 is a good time to find a group of test users that can help you evaluate the user interface.
- Phase 0 is also the time to perform Contextual Inquiry, if relevant.

Length of Phase 0

- Phase 0 can be as long or as short as necessary, it does not have to follow the rules for sprint length.
- Phase 0 can stretch over a long period of calendar time, not necessarily using so much effective time.
- Phase 0 is normally not manned by the entire Scrum team.
- There will probably be workshops with the stakeholders, so their calendars affect the length of the phase.

Sprint 0

- If you are starting a project from scratch, with no infrastructure already in place, it may be a good idea to use the first development sprint to set up the technical foundations:
  - Version control, branching strategy
  - Continuous integration, continuous delivery pipeline
  - Automation for Specification by Example tests
  - Initial system architecture
- In sprint 0, the focus is on technical issues, but you should always try to implement at least one user story, using the complete system architecture.