THE SCRUM PLAYBOOK

DOCUMENT FOR SCRUM MASTERS AND PRODUCT OWNERS WHO ARE JUST STARTING

WRITTER BY

DEJAN MAJKIC

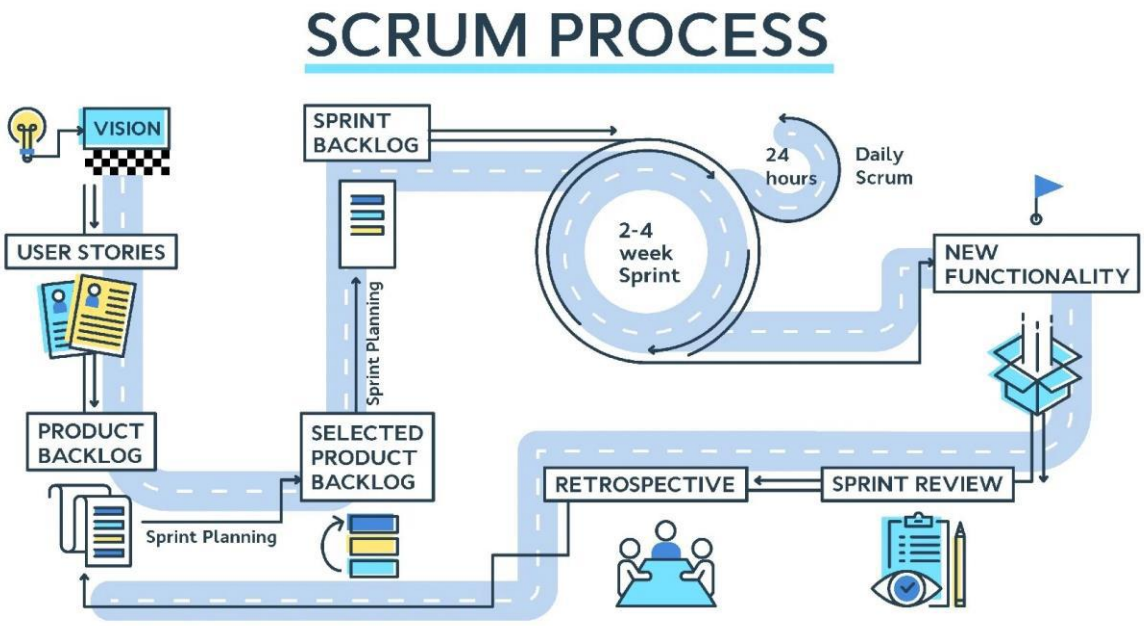
The Scrum Playbook

Contents

Introduction to the Scrum Playbook

This is not an official book, rather it's a working document for Scrum masters and Product Owners who are just starting, this Scrum Playbook is essential for reference, but their use doesn't end there. As new servant-leaders gain experience, the Scrum Playbook will remain the go-to source of inspiration for coaching and leadership.

The purpose of creating this book was to help with the human side of Scrum, rather than just another version of the Scrum Guide. I wanted insight into how to deal with the real-world problems Scrum masters face trying to help teams collaborate.



Building a successful product usually involves teams of people, and many choose the Scrum approach to aid in creating products that deliver the highest possible value. Implementing Scrum gives teams a collection of powerful ideas they can assemble to fit their needs and meet their goals.

In Scrum, we don't waste time,

Let's get started…

- Scrum deals with complex work - new product development
- It is impossible to predict from the beginning how things will be developed

- Scrum imposes timeboxes and feedback loops, and deals with uncertainty
- Sprint
    - Combines all aspects of the work
    - A scrum team will try to create a product increment
    - After each sprint, demonstrate accomplishments and discuss the next steps
    - Short iterations allow for constant feedback and improvement so that the probability of creating the “right” product is higher
- Scrum Team has three stakeholders
    - Product owner
        - Creates features, called the product backlog
        - Most important is at the top
    - Scrum master
    - Developers
- Product Backlog
    - The most important features are put at the top
    - Developers will try to take a set of high-priority features and organize them into a sprint
- Daily Scrum
    - Keeps work moving forward, identifies problems
- Scrum Master
    - Keeps team focused
- Product/Sprint Increment
    - By the end of the sprint, there should be some type of deliverable product
    - Stakeholders then evaluate the process and conduct a retrospective look at the sprint
    - Then, the next sprint begins

SCRUM Artifacts

- Three artifacts
    - Product Backlog
    - Sprint backlog
    - Product increment
- Artifacts provide
    - Transparency
    - Inspection
    - Adaptation
- If artifacts are not transparent enough, then the decisions that are made will be flawed
- The scrum master must work with everybody within and outside of the scrum team to ensure that artifacts are transparent
- Transparency involves:
    - Coaching
    - Learning
    - Understanding that change does not happen overnight

Product Backlog

- Made up of an ordered list of everything that is needed in the product
    - Features
    - Improvements
    - Fixes
- As long as the product exists, so will the backlog
- If multiple teams work on the same backlog, there should only be one backlog
- Items in Backlog:
    - Description
    - Order
    - Estimate
    - Value
    - Test description - tests, if the work performed, is complete
- The Product Owner is responsible for the backlog, from content to order
    - The Product Owner will try to understand and develop the backlog by maintaining a close relationship with the stakeholders
    - The Product Owner will add features and items to the backlog as they see fit
    - The Product Owner must closely work with developers
- Product Backlog Refinement
    - Meeting to add details, estimates, and order to the backlog
    - Developers are responsible for all estimates
    - Should not occupy more than 10% of the capacity of the developers
    - Grows with the product itself
    - Changes based on business requirements, marketing conditions, or any other relevant factors
- Most important items are put first and add more value to the product in general
    - It is OK for the lower items to have a lower degree of specificity and description
    - Higher priority items are refined and should be accomplished in the sprint
    - Items at the top should be **ready** for the next sprint
        - Should be small and detailed enough to be accomplished during the sprint
    - A product backlog item is finished when the team determines that it can be done

Example of a Product Backlog (Local Farmstore Example)

- Order
    - The first item may be the “Home Page”
    - Then, Display One Product
    - Order Form
    - Display Multiple Products
    - Add to Cart

    - Credit Card payment
- Every Product Backlog Item must have:
    - Description
    - Order
    - Estimate
    - Value
    - Test description
- User Story - in practice
    - Backlog items are sometimes called stories
    - User stories are short descriptions of a feature explained from the point of view of the person who wants the new feature.
    - Example: “As a user, I want some feature for XYZ reasons.”
    - Not every backlog item has to be in the form of a user story
    - There is no template for creating a “user story” backlog
- Create Home Page User Story
    - Description: As a customer interested in this farm store, I want to be able to open my browser and view the basic information about the company so that I can contact someone in case I have questions
    - Test Description: I should see the logo of the farm store. I should see the contact information
- Display One Product
    - Description: As a customer interested in the farm store, I want to be able to open the browser and find some basic information about XYZ products so that I am better informed about the ingredients and available sizes
    - Test Description: The XYZ product is the best-selling product and many customers ask about it.
        - The home page will include a button called “Our Products”
        - The Product Page should show
            - One or more images of the product
            - A short description
            - FAQ section
- A product backlog is not complete, nor will it ever be
    - More items will be added and defined in further sprints
    - The development can also write backlog items and user stories
        - The product owner is still responsible for this product backlog

How to Manage the Product Backlog

- An initial product backlog is just enough to get started
- Most companies use software to create product backlogs - for example, JIRA
- Scrum Guide does not provide a specific way of handling the product backlog
    - This can be done with notes, on a whiteboard, etc.
- Anyone on the team should be able to view the product backlog
    - This makes it **TRANSPARENT**

Product Backlog Refinement

- The next step is to take the user story and clarify the details and order and get an **estimate**
    - This happens during the product backlog refinement
- Scrum Master will explain the ESTIMATE
    - An estimate is the best guess of the effort necessary to accomplish a given task
    - It is NOT a commitment - there will be some uncertainty
    - Scrum is different from our PM techniques
    - To build functionality, you want to understand the scope and complexity of the task
    - Planning Poker
        - Gives you an estimated range that has larger increments
        - 0,1,2,3,5,8,13,20,40,100
          Your developers can only choose from these numbers
    - The scrum master will ask each member how they arrived at the estimated number
    - Team members will then discuss
    - You can then do another round of estimates
    - This is when the product owner will make adjustments based on recommendations
    - The estimate will be conducted again, after these changes

How to Add Estimates and Values to the Product Backlog Items/User Stories

- The product owner can now introduce the estimate to the product backlog
- The owner will now introduce the estimate by giving several story points, for example
- Value
    - Scrum Guide is not specific about this
    - Scrum mandates that each product backlog item should have a value
- Value Scale
    - Starting from 100 to 1000
    - The Product Owner can create different backlog items and assign them to this scale
    - Value is **just a guess**, it has not been measured
        - The product owner must measure the decision of the product backlog item to see whether it led to the expected results
    - The value should be transparent
    - Nothing should be done without creating value

Sprint Backlog

- During the sprint planning meeting, the product owner comes with an objective that the sprint should accomplish
- Steps

- The scrum team formulates a sprint goal
- The product owner will explain which product backlog items will help achieve the sprint goal
- Developers will decide which items from the backlog will be added to the sprint
- The sprint backlog will have a plan for realizing the sprint goal

Difference between Product Backlog and Sprint Backlog

- A product backlog is an ordered list of ideas and features that the product should/could have
- Sprint backlog contains the product backlog items for the current sprint and a plan for how to deliver these goals
  - What - are the goals for the sprint
  - How - how will we accomplish this
  - Forecast - developers will estimate which items they can deliver within the sprint
  - Process Improvement - should at least have one improvement
  - A sprint log is a TEMPORARY artifact that exists only during the sprint
    - This is the responsibility of developers
    - They OWN the sprint backlog, and nobody but them can change them
  - The sprint backlog breaks down each product backlog item into smaller units of work that allow the team to build an increment
  - The total amount of work remaining in the sprint backlog will be tracked and updated at least once a day in the scrum meeting
  - Developers are responsible for monitoring progress and the likelihood of achieving the sprint goal

How to Create a Sprint Backlog from Product Backlog Items?

- Developers will choose which items they will accomplish in the sprint, as well as the length of the sprint - they will select from the TOP of the product backlog
  - The product owner decides the priority of the product backlog items, NOT developers
- In our farm store example, we would create the first sprint
  - Show the most popular product on the website
  - This creates the temporary artifact - Sprint Backlog
  - It will only live as long as the sprint backlog lives
  - We can now call these goals Spring Backlog Items
- Developers will now start to break down the Spring Backlog Items
  - How do we implement this?

    - Tasks are attached to the Spring Backlog
    - Examples of some tasks
        - Create a git repository
        - Create a simple HTML page
        - Display Contact Information
        - Create a CD/CI Pipeline***
        - Setup a hosting account
        - Test the website on multiple browsers
    - Developers can still add additional items to the spring backlog

Increment and the “Definition of Done”

- The increment is represented by all product backlog items completed during a sprint plus the value of all increments from all previous sprints
- It is a “building block” on top of previous sprints
- Each new increment is a **useable** and **improved** version of the product
- By the end of each sprint, the increment must be complete by the “definition of done”
- Definition of Done
    - Everyone must understand what “done” means
    - The development must create a definition of done
        - Each sprint will create an increment that must adhere to this definition
    - If a sprint backlog item does NOT follow the definition of done, it will not be included in the product increment
- Quality should not be decreased, and the definition of done should NOT be changed during the sprint
- The definition of done should evolve and increase product quality as time goes on
- There is no standard for the definition of done, but it creates transparency for everyone involved
- The definition will also guide developers in choosing HOW MANY product backlog items will be included in the sprint
- The definition of done is NOT classified as an **artifact**

Example of Definition of Done

- A Product Backlog Item must meet these criteria:
    - Unit tests are written and passing
    - Acceptance tests are written and passing
    - Accessibility testing using https://wave/webaim.org/
    - The continuous Integration (CI) pipeline is passing
    - Desktop cross-browser testing is done on at least the main browsers
        - Chrome
        - Firefox
        - Safari

        - Microsoft Edge
    - Mobile cross-device testing is done on the following devices
        - iPhone, Samsung, Google
    - Performance
        - Each transaction should be completed within 2 seconds (specific interval)
    - Availability
        - At peak times, the platform should support 5,000 concurrent transactions
    - Security
        - All external code libraries must go through a security testing tool that ensures that there are no known security vulnerabilities
    - Usability
        - The platform must support visually impaired customers and follow best practices
    - Code was reviewed by two independent developers who did not work on the code
    - Technical documentation was written and peer-reviewed

Definition of “Ready”

- **Not part of Scrum Guide
- Definition of Ready refers to Product Backlog Items still in the Product Backlog and will be selected for the coming Spring Backlog by developers
- Teams use the INVEST principle - a Product Backlog Item should be:
    - **Independent** of all other Product Backlog Items or other teams/external factors
    - **Negotiable** - not a specific set of features to make room for negotiations between the Product Owner and Developers as more is learned
    - **Valuable** - should provide business or any other type of value
    - **Estimable** - the developers should have no issues making an estimation
    - **Small** - should fit within a sprint
    - **Testable** - should be clear what the outcome should be, we can test to predict this outcome

Scrum Events

- Creates a routine
- Events
    - Sprint - contains all events
    - All events are time-boxed (maximum duration)
    - All events are transparent and allow for inspection
    - Sprint Planning
    - Daily Scrum - every day of the sprint
    - Sprint Review - held at the end of the sprint to review the Increment
    - Sprint Retrospective - improve all events and adapt

Sprint

- Has a timebox of one month or less for a shippable product feature
- If the sprint is too long, the **complexity** and **risk** may increase
- Having short horizons makes it easier to plan what is built and get early feedback
- Sprints contain all Scrum Events
    - A plan on how to build a Product Increment
    - Outlines the development work needed
- Sprint Goal - will be met within the Sprint Timebox, and helps developers understand why it is building the increment
- No changes should be made that will alter the Sprint Goal
- The scope of the sprint can be clarified and renegotiated
- A new sprint starts immediately after the sprint has been accomplished
- Nothing happens between the sprints and there is no gap

Sprint Planning

- Sprint Planning is timeboxed (should not exceed) **8 hours** for a one-month Sprint
- For small sprints, it will be shorter
- During Sprint Planning, the Product Owner and Developers will agree on a sprint goal and discuss which items will be in the Sprint Backlog
- The work that needs to be done will be decided during Sprint Planning
- Steps
    - 1. What can be built - product owner presents the goal of the sprint and the backlog items that encompass this
        - Developers forecast what can be done
            - The number of Product Backlog Items is selected by the Developer Team; the Product Owner cannot influence this decision
        - Scrum Team must take into account
            - What was accomplished during the previous increment
            - The projected capacity of the developer team
            - The past performance of the developer team
    - 2. How to Build It
        - When needed, developers can invite others from outside of the team who provide technical or domain advice and perspective
        - The work plan for the first day of the sprint is broken down into units of one day or less by the end of this meeting
        - Because work merges during the sprint, this meeting cannot identify all work needed during the sprint
            - It is just a plan so that development can start
        - Developers should be able to explain to the Product Owner and Scrum Master their plan and their forecasted increment

Daily Scrum

- A timeboxed event held at the same time and place each day to reduce complexity
- Held every day of the sprint
- It is an event made for developers
- Developers plan what will be done in the next 24 hours
- Key “Inspect and Adapt” Meeting
    - Helps development inspect progress
- The Daily Scrum structure is set by the developers
    - Can be questions or more discussion based
- Scrum Guide Gives Question Examples
    - What did I do yesterday?
    - What will I do today?
    - Do I see any potential issues in the way?
- Regardless of the size of the Scrum Team, the Daily Scrum is a maximum of **15 minutes**
- The Scrum Master must make sure that the Team is having the meeting, and that it is maintaining the timebox of 15 minutes
    - The Scrum Master must ensure that any outside contributors are not interrupting the meeting if they are present
    - The Daily Scrum is NOT designed to explain the progress to the Product Owner or Stakeholders
    - Daily Scrums improve communication, identify impediments, and help the team to make quick decisions

Scrum Board

- Each developer will be given a specific task
- Everyone will know which tasks are at which stage (in progress, completed, etc)
- This is how the developers communicate with each other
- This is used during the Daily Scrum

Sprint Review (Meeting)

- At the end of the Sprint, a Sprint Review is conducted
- By the end of the Sprint, there should be a shippable product
- The Sprint Review inspects the increment and adjusts the Product Backlog if needed
- The **Product Owner** owns this meeting and will invite other stakeholders to this event
- Scrum Master facilitates this meeting and ensures that it is held within the timebox
    - This will be a maximum of **4 hours** for a one-month sprint
- This is an informal meeting
- The analysis of the increment is done to receive feedback and encourage collaboration on the next steps

- Typical Agenda
    - Which Product Backlog Items have been done, and what has not to be done
    - The Product Owner should then explain how the Product Backlog Items add **value**
    - The highlight of the meeting is the **demonstration** by Developers of the Product Backlog Items that have been accomplished
        - Stakeholders should “try out” the product
        - Developers also demonstrate what progress has been made, issues that arose, and how they were solved
    - The Product Owner discusses the Product Backlog and explains when the next release will be available
    - This meeting also evaluates any changes in the market or other factors that will impact future development
        - This is an opportunity for stakeholders to ask questions or suggest changes for new features to the Product Owner
    - Product Backlog is revised
    - ONLY completed Product Backlog Items that have been completed
        - They are NOT part of the Increment

Sprint Retrospective

- The last event of the Sprint, after the Sprint Review meeting
- This focuses on inspecting and adapting the **process**
- During this meeting, the team should **inspect** and **adapt** for the next sprint
- This is an internal Scrum Team Event - no stakeholders or Product Owner
    - Outside feedback should occur outside of this meeting
- The Scrum Master facilitates this meeting, makes sure that everyone understands the purpose, ensures that the meeting is positive, and keeps it within the timebox
- For a one-month sprint, the maximum time limit is 3-hours
- Example Meeting Structure - How was the last Sprint
    - People
    - Relationships
    - Processes
    - Tools
- Identify the major items that went well and identify improvement
- Create a plan for implementing these improvements
    - May update “Definition of Done”
        - Will include quality criteria
    - Ensure transparency

Sprint Retrospective in Practice

- Scrum Guide does not explain the details of this meeting
- Every Scrum Retrospective can be different, as long as it adheres to these rules

- Scrum Master plans and organizes this meeting
- It is recommended to have a retrospective where everyone is physically present
- Use pens and post-it notes
- Process
    - Warm Up - an icebreaker, kick off the meeting, energize the team
    - Check-in - how is everybody feeling?
    - Gathering Data - what helped the sprint, and what did not?
    - Individual Reflection - each member has a certain timebox to reflect on the sprint and write down ideas
        - After the timebox finishes, each team member will share
    - Sorting - create clusters of issues/topics, and organize them by importance
        - There is not always enough time to discuss everything
        - Dot Voting - each team member has a fixed number of dots, and each member votes for what they think is the most important issue
            - The issues with the most dots will be discussed in further detail
    - How was the sprint?
        - Discuss the most important issues and improvements
        - Create action points and next steps

Canceling a Sprint

- Is **very rare**
- A Sprint can be canceled if:
    - The Sprint Goal becomes obsolete
    - Major changes in the market occur
- Because Sprints are very short, this is very uncommon
- Only the **Product Owner** has the authority to cancel the Sprint
- If a Sprint is canceled:
    - **Completed** Product Backlog Items will be reviewed
    - Incomplete Product Backlog Items will be re-estimated and **put back** in the Product Backlog

The Scrum Team

- Consists of:
    - Product Owner
    - Developers
    - Scrum Master
- Scrum Teams are:
    - Self-organizing
        - Choose how to best accomplish work; they decided **how** the product should be built
    - Cross-functional

        - A team that has all of the skills needed to accomplish the work without depending on people outside of the team
- Scrum maximizes:
    - Creativity
    - Flexibility
    - Productivity
- Develops products in an incremental way

Product Owner

- The main responsibility of the Product Owner is to maximize the value of the product resulting from the work of the developers
- What is value? (**Not specified in Scrum Guide)
    - Business value - more income, better exposure in the market
    - Value for customers - better value for the customers
    - Technical value
- This is ONE person
    - Can represent the desires of stakeholders (CEO, company, etc.)
    - Product Owner decides the priority of Product Backlog
- This person is solely responsible for the outcome
    - This is why the Product Owner is one person - their decisions should be absolute
- ***For the Product Owner to succeed, the entire organization must respect his/her decision
- Everything must be transparent - all decisions made by the Product Owner, what the developers will be working on next
- Product Owner must make sure that the Product Backlog Items are clearly expressed and understood by the Developers
- ***The Product Owner remains accountable, even if he/she delegates tasks to the Developers
    - No one can force the Product Owner to delegate responsibilities
- In Scrum, **there is NO Project Manager**
    - **No one is reporting to, or giving updates to, the Product Owner**

Developers

- Consists of specialists who have ALL of the skills to do the work needed
- Goal: to produce a shippable increment with each sprint
- *Only developers can work on the Product Increment
- The team is empowered by the organization to organize their work
- Characteristics
    - Self-organizing - nobody, not even the Scrum Master, Product Owner, or CEO can tell them how to do anything
        - Applies to problem-solving and decision-making as well
    - Cross-functional - has all of the skills needed to complete the task

        - Not every member is cross-functional, just the team as a whole
- The accountability for the work belongs to the team as a **whole**
    - If a developer is sick and something is not completed, this falls on the entire team, not just the sick developer
- *Scrum Guide does not recognize any official titles
- Team Size
    - Fewer than 3 developers **decrease interaction and result in lower productivity**
    - Additionally, with fewer developers, there will be **skill constraints**, meaning they will be unable to ship a Product Increment
    - Having more than 9 members **increases the complexity**
    - **The team size should be between 3 and 9 members**
        - Product Owner and Scrum Master are not included in this number
            - **unless one or the other is a part of the developers, which is not recommended

Scrum Master

- Assists Developers and Product Owner and is responsible for promoting Scrum throughout the organization
    - Helps to understand the Scrum theory, practices, rules, and values
    - Helps those outside the Scrum Team (i.e. stakeholders) to understand which interactions with the Scrum Team are helpful, and which aren't
- The Scrum Master is a skilled Servant Leader
    - He/she puts the team ahead of him/herself
    - Servant Leader
        - Turns the power pyramid upside down - puts the needs of the others first, and helps others perform as much as possible.
- The Scrum Master has **no formal authority**
    - They are not a manager
    - They are a facilitator and a coach

Scrum Master and Developers

- How do they help the developers?
    - Coaches in self-organization and cross-functionality
        - Helping organizations adopt Scrum
    - Helps to create high-value products
    - Remove impediments
        - I will work with people inside the organization to solve these
    - **Facilitating** Scrum events
        - The Scrum Master is **NOT** the secretary of the team
        - They just make sure that the meetings are happening and adhere to Scrum principles

The Scrum Master and the Product Owner

- Coaches the Product Owner to understand Scrum and Agile practices
- Ensures that goals, scope, and product domain are understood
    - If any of these are unclear or missing, there will be problems
- Scrum Master makes sure that Backlog items are clear
- Scrum Master helps Product Owner to learn effective Product Backlog management
- Makes sure that the Product Owner knows how to do product planning in an **empirical environment**
- Can also facilitate planning and Backlog refinement as requested by the Product Owner

Scrum Master and the Organization

- Coaches the organization in Scrum adoption and planning
    - Can collaborate with different departments
    - Explain why this change is necessary
- Work with other Scrum Masters
- Cause changes that increase the productivity of the Scrum Team
- Coaches Scrum and empirical product development

Scrum and Agile Methods

- What is Agile?
    - Comes from the Manifesto for Agile Software Development
        - Scrum
        - Extreme programming
        - Kanban

- Values of Agile
    - Individuals and interactions over processes
        -
    - Working software over comprehensive documentation
    - Customer collaboration over contract negotiation
    - Responding to change by following a plan
    - "That is, while there is value in the items on the right, we value the items on the left more."

- OVER doesn't imply that the other things aren't important, but that they should not be obstacles to accomplishing the goal of developing software
- The approach and plan should be flexible during the project - to be AGILE
- Scrum existed before Agile

    - Scrum was used in product development in 1986
    - Ken Schwaber and Jeff Sutherland presented a paper in 1995 on the “Scrum Framework”

What is Scrum?

- Scrum is a process framework used to manage work on complex products
    - It provides rules, it provides structure, but it is not a **solution**
- Using the framework helps people to address complex, adaptive problems, while productively and creatively delivering products of the highest possible value
- **Scrum is NOT a process, technique, or definitive method (or methodology)
- Scrum is:
    - Lightweight
    - Simple to understand
    - Difficult to master
- The Scrum framework consists of Scrum Teams and their associated roles, events, artifacts, and rules

Waterfall

- Waterfall Model
    - Originated in manufacturing and construction
    - Has been used for software since the 1950s
    - Linear-sequential lifecycle model
- Model
    - Each phase must be completed before the first can begin
    - There is no overlapping between the phases
    - Progress flows from top to bottom
- Components
    - Requirements - product requirements
    - Design - software architecture
    - Implementation - software
    - Verification
    - Maintenance
- Assumes that problems are identified and corrected in the early stages
- Downsides
    - We rarely have perfect knowledge in the beginning
    - Working software is produced late
    - Risk and uncertainty are relatively high
    - Not ideal if the premises are changing often

Uses of Scrum

- Used to develop and sustain products and enhancements
- Sustain and renew products
- Research and identify viable markets, technologies, and product capabilities

Scrum has been used for solving problems with a small group of people for

- Developing software/hardware
- Building autonomous vehicles
- In schools and governments
- Marketing

Note: Scrum is not ONLY for software development

Scrum Theory

- Scrum is founded on empiricism (control theory)
- Empiricism asserts that knowledge comes from **experience,** it is based on what has happened in the past
- Three Pillars of the Empirical Control Process
    - **Transparency**
        - Significant aspects of the process must be **visible** to those responsible for the outcome
        - Observers need to have a common understanding of what is being seen
    - **Inspection**
        - Scrum users must regularly inspect Scrum Artifacts
            - Product Backlog, Sprint Backlog, Increment
        - Scrum users must progress toward a Sprint Goal to detect undesirable results
    - **Adaptation**
        - As soon as a deviation from the expected result is detected, an adjustment must be made as soon as possible
        - Scrum prescribes four formal events for inspection and adaptation
            - Sprint Planning
            - Daily Scrum
            - Sprint Review
            - Sprint Retrospective
- You must regularly inspect what you're doing to determine whether what you are building is on track
    - If it is not on track, you must make an immediate change

Scrum Values = Trust between Scrum Team Members

- Commitment
- Courage
- Focus
- Openness
- Respect

Professional Scrum Competencies

Understanding and Applying the Scrum Framework

- Allows teams to iteratively and incrementally deliver valuable products of “Done,” working, releasable software in 30 days or less
- Scaling - in the case of scaled implementations of Scrum, minimizing cross-team dependencies and resolving integration issues are unique and critical challenges when multiple Scrum Teams are collaborating to deliver a product

Key Focus Areas

- Empiricism - a practitioner of Scrum will be able to apply the concepts of the empirical process to the problems that they encounter - apply scientific methods to complex problems, and understand why and how to apply an empirical process
    - Describe problems in terms of learning
    - Break problems down into the smallest increments that generate valuable evidence
    - Execute in an empirical way
- Scrum Values
        - Focus
        - Respect
        - Openness
        - Commitment
        - Courage
    - A practitioner should be able to apply them in organizations whose values do not match Scrum
    - This leads to an empirical process, self-organization, and continual improvement (CI)
- Roles
    - Product Owner
    - Scrum Master
    - Developers
- Events
        - The Sprint
        - Sprint Planning
        - Daily Scrum

        - Sprint Review
        - Sprint Retrospective
    - The events are used to uphold empirical process control through the three pillars of Scrum
        - 1. Transparency
        - 2. Inspection
        - 3. Adaptation
- Artifacts
    - The artifacts provide a team with a minimal set of materials to plan, execute, and review the Sprint
        - Product Backlog
        - Sprint Backlog
        - Increment
- “Done”
    - The objective of each Sprint is to deliver a “Done” product increment. The Definition of Done provides a way for a team to make what “Done” means transparent
- Scaling
    - Scrum is designed to work at the team, product, and organization level
    - The ultimate level of proficiency within this Focus Area is the ability to know what, and what not, to compromise in pursuit of a scaling approach by understanding the trade-offs and benefits of particular concepts and practices.

Developing People and Teams

- Self-Organizing Teams
    - Self-organizing and empowered teams are the engine to delivering value
    - Practitioners should understand how to incrementally introduce self-organization, the practices that help self-organization to occur, and the measures that help one judge whether a team can be empowered to self-organized
- Facilitation
    - Making decisions, sharing ideas, and being transparent is difficult
    - Facilitation is a set of practices that help support the collaboration, communication, and creativity of teams and individuals
- Leadership Styles
    - Practitioners should understand the concepts of leadership styles, and be able to apply a particular style when the situation calls for it, decide on the right style, and understand its impact on the organization
- Coaching and Mentoring
    - A key aspect of **Servant Leadership** is the ability to coach and mentor the organization, the team, and the business
    - The objective is to help people get better at work, deliver more value, or resolve a conflict or problem
- Teaching

- The ability to inspire others to learn and share information is effective, repeatable, and efficient
- A practitioner should appreciate the means of measuring the success of their teaching

Managing Products with Agility

- The process of aligning Product Vision with Product Value is **iterative** and **incremental,** and it is managed through continuous refinement of the Product Backlog.
- The Product Owner is typically focused on Managing Products with Agility, but this competency should be shared amongst all Scrum Team members, Agile Leaders, and organizational stakeholders

Focus Areas

- Forecasting and Release Planning
  - Practitioners should understand how releases should be planned while dealing with complexity, dependencies, and value creation
- Product Vision
  - Defines the purpose or goal that the product serves, and is defined by the "value" that the product strives to deliver
  - It should be the "true north" for the product, and should not be affected by day-to-day difficulties
  - The Product Vision only changes if the goal of the product changes, such as when a business pivot happens
  - Scrum Masters should be able to describe the vision, and which techniques should be employed to both build a vision and make it transparent
  - Practitioners should also use the Product Vision to drive strategy and execution, and how to build a vision that motivates, communicates, and provides constraints for delivery
- Value
  - The ultimate goal is to deliver value to the customer and the stakeholders
  - Practitioners should be able to understand how to define value for context and apply it to the work that they and the team do
  - They should be able to manage others' understanding of the value and apply different techniques and practices for defining, communicating, and measuring value
  - They should understand the connection between value and the empirical process, and how value should be the driving factor of the Product Vision
- Product Backlog Management
  - The **Product Backlog** is the key **artifact** within Scrum.
  - It provides transparency into what is happening to the product for the team, organization, and stakeholders
  - The practitioner should be able to describe what a Product Backlog is, and apply a variety of techniques for managing the backlog. Also, how to make a Backlog transparent, and manage stakeholder expectations associated with the Backlog

- Business Strategy
    - A product lives within the context of a business strategy
    - The strategy describes how the Product Vision will be executed in a broader context
    - A practitioner understands techniques for exposing business strategy and showing how it drives the product
    - Practitioners should understand **Lean Startup** and **Design Thinking**
- Stakeholders and Customers
    - Effectively working with stakeholders is key for both the Product Owner and the Developers
    - Scrum encourages **more frequent collaboration** and more **open dialogue**

Evolving the Agile Organization

- Evolving the Agile Organization includes concepts and tools for measuring and enabling business agility through **Evidence-Based Management (EBM)**
- **EBM** encourages rational, data-driven decisions when applying empirical process theory to the development of high-value products
    - It includes the use of three key measures
        - 1. **Current Product Value**
        - 2. **Time to Market**
        - 3. An organization's **Ability to Innovate**

Key Focus Areas

- Organizational Design and Culture
    - Complex problems require a different way of organizing
    - The fundamental differences of an agile organization are:
        - 1. Structure
        - 2. Design
        - 3. Culture
- Portfolio Planning
    - Practitioners should understand why agile portfolio planning must be different than traditional portfolio planning to deal with complex products and systems
- **EBM**
    - A fundamental element of Scrum is an empirical process: the idea that complex problems require real experience to effectively plan and deliver value
    - EBM provides a value-based approach

The Scrum Framework Main Ideas

- Scrum is a framework for developing complex products

- Scrum addresses complex problems in an **iterative** way
- Scrum is lightweight, simple to understand, **but difficult to implement**
- The Scrum framework consists of Scrum Teams and their associated roles, events, artifacts, and rules
- Scrum is used in different disciplines, from software development to operations management, as well as in day-to-day life
- The essence of Scrum is a small team of people
- Scrum is founded on **empirical process control theory** or **empiricism**
- The three pillars of empirical process control are:
    - 1. Transparency
    - 2. Inspection
    - 3. Adaptation
- The Scrum Values are:
    - 1. Commitment
    - 2. Courage
    - 3. Focus
    - 4. Openness
    - 5. Respect

Nexus Framework

- Explains how to scale Scrum - how to operate many Scrum teams at the same time

Scaling Scrum - Multiple Scrum Teams Working on the Same Product

- **1 Product = 1 Product Backlog = 1 Product Owner**
- Even if there are many Scrum teams, there will only be ONE Product Owner
- A Scrum master **can** work in multiple teams
- There will not be a separate Product Backlog for each team
    - During Sprint Planning, Developers will still pull items from the Product Backlog in agreement with the Product Owner

Dependencies Between Scrum Teams

- Key Concern: reduce the dependencies between the teams
    - This is less of a concern if there is **enough work** for everybody
    - There is a high velocity on the developers

Why is there only one Product Owner?

- Only one person is responsible
    - Having a committee can make decision-making complicated
    - Additionally, when compromises are made, the decision may not be the **best** one
    - Shared ownership makes it difficult to know who is involved, who is responsible
    - There can be finger-pointing if things go wrong
    - Micromanagement and coordination issues
- There is a clear decision-making process
- In Scrum there is no Chief Product Owner, nor is there a Proxy Product Owner

Definition of Done for Multiple Scrum Teams

- Developers must create a mutual, common Definition of Done
    - All teams must respect this
    - The Product Increment is only done when it is integrated, useable, and releasable
    - Each Scrum Team can have more stringent Definitions of Done for their work, as long as they respect the common Definition of Done that was defined and agreed upon by all teams

Aligned Sprints

- Scrum does not require that Sprints must be conducted on the same timeline
    - The Product Increment should be integrated by the end of the Sprint
        - By the end of each Sprint, the Product Increment should be integrated, even if the Sprints do not align
- If the Scrum Teams have different Sprint lengths, this can be more difficult, but it is still part of Scrum
- ****There are no Hardening Sprints or Integration Sprints in Scrum**
- Any work (integration, testing) must be done within the Sprint

Persons and Roles in Scaling Scrum

- **A person can play different roles in Scrum**
- For example, a Scrum Master can be on two teams
- If two teams are working on two different products
    - Each team will have a Product Owner and a Scrum Master
- The Scrum Guide puts definitions on the **roles,** not the individual persons

Burndown Chart

- The amount of work or number of story points vs. the time of the Sprint
- Is used to estimate when the work will be completed, and whether it will be completed on time
- The graph includes a “guideline” that indicates ideal linear progress
- If using story points, this graph will be less accurate
    - If more work is added during the sprint, the work remaining will go up, and the graph will spike
- Used to predict the developer's likelihood of completing the work on time
- **Burndown charts are **not mandated by Scrum**
    - Don’t be fooled by questions that reference burn charts as part of Scrum principles
    - It is up to the Developers which tools to use
- Burndown Charts are only related to the **remaining work** and are not related to **project costs, business value, or the productivity of the developers**

Burnup Chart

- Provides a representation of a Sprint’s completed work related to its total scope, which is represented by the number of selected Product Backlog Items (or Stories)
- In contrast with the Burndown Chart, **added work will be visible in a change of the scope**
- The work progress will start from 0 and trend towards the scope (the total work that was forecasted)
- ****Scrum does not enforce any of these tools or charts**

Technical Debt

- Scrum Guide does not mention this
- AKA Design Debt or Code Debt
- Unresolved technical aspects from the past come back to haunt you today
- Technical Debt should be continuously dealt with and not postponed
- This can accumulate from the shortcuts that the developers took in accomplishing an original project
- If the developers don’t have the time to update or fix the past problems, then the product becomes hard to maintain
- Technical Debt is tied to the quality of the product
    - The Product Owner and the developers should work together to maintain this
- How to deal with it as a Scrum Team
    - Transparency
        - Developers should communicate current issues
        - The team needs to collaborate with the Product Owner in every Sprint

        - Tasks designed to eliminate Technical Debt should be in the Product Backlog
        - Adapt your Definition of Done
- There is software that tracks code complexity and technical debt

Cone of Uncertainty

- Used in software development when the technical and business environments are constantly evolving
- At the start of the project, comparatively little is known about the Product or work results, and the uncertainty is high
- As more is learned, uncertainty tends to decrease
- Typically, an exam question may include a reference to this term

Velocity

- Velocity is a measure of the amount of work that developers can handle during a typical Sprint
- During the Sprint Planning Meeting, the developers will choose the Product Backlog Items
- Each Item will have a set of Story Points for the Sprint
- Velocity = Work Sprint 1 + Work Sprint 2 + Work Sprint 3 / number of Sprints
- In JIRA, there is a velocity report with the last 7 sprints
- ****Velocity is not mandatory in Scrum**
- Velocity is not correlated with **VALUE**
    - More output does not always equal more value delivered
- There is no commitment in Scrum! There are no velocity goals in Scrum
    - Developers should not be punished for having a velocity lower than predicted

Engineering Standards

- Refers to a set of rules and conventions within the development organization that applies to all of the Products being developed

Feature Teams vs. Component Teams

- *Not mentioned in the Scrum Guide
- Layers of the Application
    - UI
    - Server Application
    - Database
- A Feature Team works through all of the layers of the application to fulfill a customer's need
    - *DESIRABLE, but not mandatory

    - Has all of the skills needed to complete this feature
- Component (Layer) Team - focused on single or multiple components of the system
    - A Component Team does not have the skills to work across all of the layers
    - A Component Team may be focused on specifically the UI, or maintaining the database, but not all of the functionality of the software
- Using Component Teams increases dependencies between teams, and reduces the chances of producing an integrated Product Increment
    - All teams must integrate their work during the Sprint
    - If one team does not accomplish goals by the end of the Sprint, then the other teams cannot accomplish their goals either

Sustainable Pace

- **Working overtime to complete a Sprint or reach the Sprint goal is not acceptable in Scrum**
- **Scrum is about finding a good balance and keeping everyone happy.**
- Short-term gains at the expense of the happiness of the developers are not worthwhile

Functional and Non-Functional Requirements

- Functional Requirement - can be easily understood by the Product Owner, Stakeholders, or users of the product
    - Example: In our farmhouse, the ability to order a product from the website is a functional requirement
    - It is a function of the product
- Non-functional Requirement - describe qualities, behaviors, attributes, and constraints of the product
    - Can be placed in categories
        - Performance
            - Each transaction must be completed in under 2 seconds
        - Security
            - All external code libraries should go through a security testing tool
        - Availability
            - At peak times, the platform should support 5,000 concurrent transactions
        - Usability
            - The platform should support visually impaired customers, and follow best practices in the industry
    - Not so easy to grasp, it may be hard for others to understand outside of the developers
- How to manage Non-Functional Requirements
    - The Most Common Approach is to describe Non-Functional Requirements in the “Definition of Done”

    - *The Scrum Guide clearly states that the Product Backlog is the single source of requirements for any changes to be made to the product
    - Non-functional Requirements can be listed as Product Backlog Items, or as Acceptance Criteria for Product Backlog Items
- **Exam Tip:** Functional and Non-Functional Requirements are the responsibility of the **Product Owner**
    - For Non-Functional Requirements, it is common for the Product Owner to collaborate closely with developers, but they are still the responsibility of the Product Owner
- **Exam Tip:** There is no separate backlog for Non-Functional Requirements!

Hardening Sprint or Integration Sprint

- A Hardening Sprint, or Integration Sprint, is typically a Sprint in which no new features are involved.
- Some organizations use these to merge the work of multiple Developers and to test the Increment before it ships to Production
- **Exam Tip:** This practice is against the rules in the Scrum Guide
- In Scrum:
    - There is no Integration Sprint
    - There is no Hardening Sprint
    - There is no Sprint 0
    - There is no “after Sprint”
    - All Sprints are the same, and everything that happens is within a Sprint
    - **Each Sprint MUST produce a Product Increment**
- **Teams (regardless of when their Sprint ends) must collaborate and make sure that a Product Increment is created by the end of each Sprint**

Glossary

- Coherent/Coherence - the quality of the relationship between certain Product Backlog Items which may make them worthy of consideration as a whole. See also: Sprint Goal
- Emergence - the process of coming into existence or prominence of new facts or new knowledge of a fact, or knowledge of a fact becoming visibly unexpectedly.
- Forecast of Functionality - the selection of items from the Product Backlog that a developer deems feasible for implementation in a Sprint
- Increment (Scrum Artifact) - defines the complete and valuable work produced by developers during a Sprint. **The sum of all Increments forms a Product.**
- Ready - a shared understanding by the Product Owner and developers regarding the preferred level of description of Product Backlog Items introduced at Sprint Planning
- Stakeholder - a person external to the Scrum Team with a specific interest in and knowledge of a product that is required for incremental discovery. Represented by the Product Owner and actively engaged with the Scrum Team at **Sprint Review**

- Technical Debt - the typically unpredictable overhead of maintaining the product, often caused by less-than-ideal design decisions, contributing to the total cost of ownership. May exist unintentionally in the Increment or be introduced purposefully to realize value earlier
- Velocity - an optional, but often used, an indication of the amount of Product Backlog turned into an Increment of product during a Sprint by the Scrum Team, tracked by developers for use within the Scrum Team
- Values - When the values of Commitment, Courage, Focus, Openness, and Respect are embodied by the Scrum Team, the three pillars (Transparency, Inspection, and Adaptation) come to life and build **trust** for everyone. The Scrum Team explore those values as they work with the Scrum events, roles, and artifacts

Definition of Done

- At which stage is it created?
    - This is not explicitly stated in the Scrum Guide
    - This is left for the Scrum Team to decide
    - Typically, a (generalized) Definition of Done may already exist within the Development Organization
- Who creates it?
    - 1. The Development Organization
        - Most organizations already have development processes in place, standards that must be followed, procedures, and quality standards. This may not be a fully-fledged DoD, but it is close
    - 2. Developers
        - Developers of the Scrum Team must define a Definition of Done appropriate for the Product, if there isn't one defined by the development organization
        - It is not explicitly stated in the Scrum Guide, but the Product Owner may be involved in this process and may help to define rules and quality criteria that are stricter, but still follow the Definition of Done that the organization has
- During which Scrum Event is the DoD discussed and finalized?
    - The DoD is **never final.** The Sprint Retrospective is used for adapting the Definition of Done, but this may happen anytime. There is no need for a specific event to do so. The team is **self-organizing.**
- Can you Change the DoD during a Sprint?
    - **Exam Tip:** There is nothing prohibiting this in the Scrum Guide.
    - The idea is to make the DoD more detailed/strict when it comes to quality
    - The DoD should never be lowered or reduced
    - **Exam Tip:** The Scrum Guide only mentions the Sprint Retrospective as an opportunity to adapt the DoD
- **Note: There is no "outside of the Spring". Everything happens within a Sprint

Mental Note: Read the Scrum Guide!

Exam Tips

- Be aware of the time
    - You WILL run out of time
    - Limit the time for the mock tests
    - Try to do 80 questions in 40 minutes
- You CAN mark questions for review
    - You may not be able to review more than 5-7 questions
- The website is SLOW
    - Keep this in mind for the time
- Keep materials near you - Scrum Guide, water, notes
- Do it when you are FRESH
- Read the questions carefully
    - There are tricks!
    - Read every answer
    - Words like “can” or “may” can change the meaning of the question
- Keep a rate of 20 questions every 10-11 minutes (1 question every 30 seconds)
- Use process of elimination

Tasty Tidbits from a Quick Read of the Scrum Guide

- How can the developers support a good application architecture?
    - Having guidelines in place which every team member knows and follows
    - Constantly working toward improving the architecture during each Sprint
- Should team members external to the developers attend the **Sprint Retrospective?**
    - Yes
    - The only Scrum Team members outside of the developers are the Product Owner and the Scrum Master and they should attend the Sprint Retrospective. So the entire Scrum Team should attend the Sprint Retrospective.
- What is the aim of the Definition of Done
    - Creates a common understanding of what completed work means
    - Increases transparency
    - Guides the developers in knowing how many product backlog items it can select for the Sprint Backlog
- When do developers collaborate with the Product Owner in the Product Backlog Refinement meeting?
    - As needed, during the sprint
    - The Scrum Team decides when (and how) refinement is done, not only the Product Owner. The Sprint is the container for everything that happens. There is nothing between two Sprints.

- The Product Owner seeks advice from the Scrum Master on forecasting when the work will be completed for reaching different business goals. How should the Scrum Master react?
    - Teach the product owner to track the total work remaining at least every Sprint Review
    - As the Product Owner is responsible for reaching goals, it is also responsible for tracking the progress toward the goals. While the burn-down flows are one way to do this, it is not the only option nor is it mandatory to use them.
- What should the length of a Sprint be?
    - One month or less
    - Short enough to keep the risk acceptable
    - Short enough to align the development work with business situations
- What are the responsibilities of the developers?
    - Monitoring productivity
    - Estimating product backlog items
    - Designing architecture
        - Developers DOES NOT facilitate scrum events
- What is discussed during the Sprint Retrospective
    - Adapting the Definition of Done
    - Working relationships
    - How the team performed during the previous Sprint
        - Note: As there are dedicated meetings for discussing and adapting the Product and Sprint Backlog, it is not appropriate to discuss it during the Sprint Retrospective.
- Who must ensure that Product Backlog Items are built according to the Definition of Done?
    - Developers
- What are the two responsibilities of the Lead Developer in the developers
    - Scrum does not recognize any titles or roles within developers
    - **Everyone** in the developers shares the responsibility of building the product

- Developers should have a working Product Increment before the Sprint Review Meeting
    - True!

- During the Sprint Planning meeting, developers notices that their capacity is smaller than the work selected for the Sprint. What are two possible actions?
    - Inform the Product Owner, start the Sprint, and closely monitor the progress toward the Sprint Goal
    - Collaborate with the Product Owner and remove or change the selected Product Backlog Items

- Who is responsible for monitoring progress towards high-level goals?
    - The Product Owner
- In which two meetings can people outside of the Scrum Team participate?
    - Sprint Review
    - Spring Planning
- Who is responsible for crafting the Sprint Goal at the Sprint Planning?
    - The **Scrum Team**
- Who is allowed to participate in the Daily Scrum?
    - The developers ONLY
    - If others are present, the Scrum Master makes sure that they do not interrupt
- What should be taken into account for the Definition of Done?
    - Conventions, standards, and guidelines of the organization
    - Definition of Done from other Scrum Teams working on the SAME product
- Who participates in Sprint Planning?
    - The entire Scrum Team works to create a plan

Your Next Steps?

Agile and Scrum Masterclass is The First and Only Course That Tells You Exactly How to Start Scrum Career.

START FROM HERE: https://www.whatisscrum.org/

Scrum was a problem I struggled with

Until I found these secret formulas.

This is how I work with Scrum now !!!