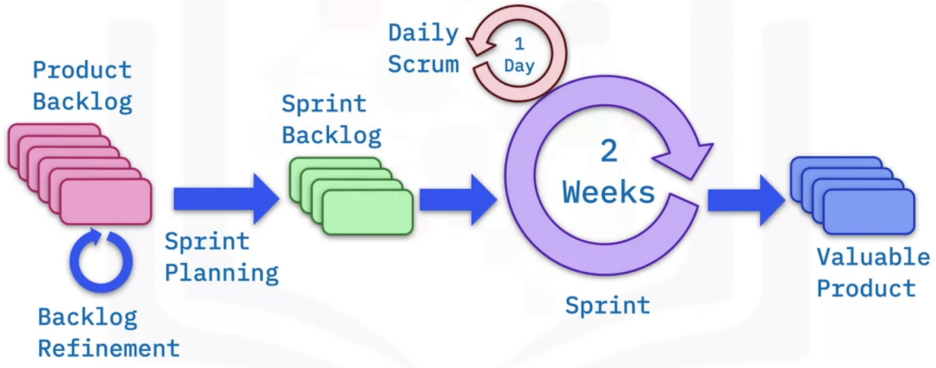
Rashid Wassan

bytewise.

DevOps Track - Week 5

Please explore the following (estimated time required: 15 hours):

1. What is Kanban and Kaban Board?
2. What are the core principles of Kanban?
3. What is Scrum? What are the different roles in the scrum (Product Owner, Scrum Master, and Scrum Team)? Explore the steps involved in a scrum process.



4. What are the benefits of scrum?
5. What are user stories and story points? What are INVEST characteristics of a story? What is epic?
6. What is a sprint? How long should it be?
7. What is a product backlog? What are effective ways to build a product backlog? What is backlog refinement?
8. Explore and get yourself comfortable with GitHub repositories, issues, and labels. Explore typical GitHub workflow with ZenHub for project management. Get familiar with ZenHub as well. https://app.zenhub.com/
9. What are sprint planning and sprint milestones? How to create user stories and add them to milestones on ZenHub?
10. What is a daily workflow in agile development? What is a daily standup? What is sprint retrospective and sprint reviews? Who are the potential attendees of these meetings?
11. How to measure sprint success? What are burndown charts?

Tasks - Week 5:

1. Get a solid understanding of these concepts and connect the dots to understand how things go on in the agile development model.
2. Create or mimic a workflow of any project sprint and implement it on ZenHub. Share the screenshots of your project board and learning on LinkedIn.
3. Write two different articles on any two topics listed above.
4. Prepare notes for all the topics in your GitHub repo.

Rashid Wassan

Please find these notes if you feel stuck in connecting the dots.

Introduction to Agile Philosophy

What is agile?

- Iterative approach.
- **Agile emphasizes:**
  - Adaptive planning.
  - Evolutionary development.
  - Early delivery.
  - Continual improvement.
  - Responsiveness to change.

Agile Manifesto

- Manifesto for Agile Software Development (agilemanifesto.org)

Agile Software Development

- An iterative approach towards Software Development consistent with the agile manifesto.
- Emphasizes flexibility, interactivity, and a high level of transparency.
- Uses small, co-located, cross-functional, and self-organizing teams.
- Build what is needed, not what was planned.

Waterfall Model Revisited

- Requirements -> Design -> Code -> Integration -> Test
- **Problems with the Waterfall Approach:**
  - No provisions for changing requirements.
  - No idea if it works until the end.
  - Each step ends when the next one begins.
  - Mistakes found later on are expensive to fix.
  - Long lead times.
  - Teams are working separately, unaware of their impact on each other.

Extreme Programming

- One of the first agile methods as it is an iterative approach.
- **Extreme Programming Values:**
  - Simplicity
  - Communication
  - Feedback
  - Respect
  - Courage

What is Kanban?

- A Japanese manufacturing system in which the supply of components is regulated through the use of an instructions card sent along the production line.

- A Kanban system is characterized by visualizing workflow, limiting work in progress, managing and enhancing flow, making process policies explicit, and continuously improving a process
- **Core Principles of Kanban:**
    - Visualize the workflow.
    - Limit Work in Progress.
    - Manage and enhance the flow.
    - Make process policies explicit.
    - Continuously improve.

Introduction to Scrum Methodology

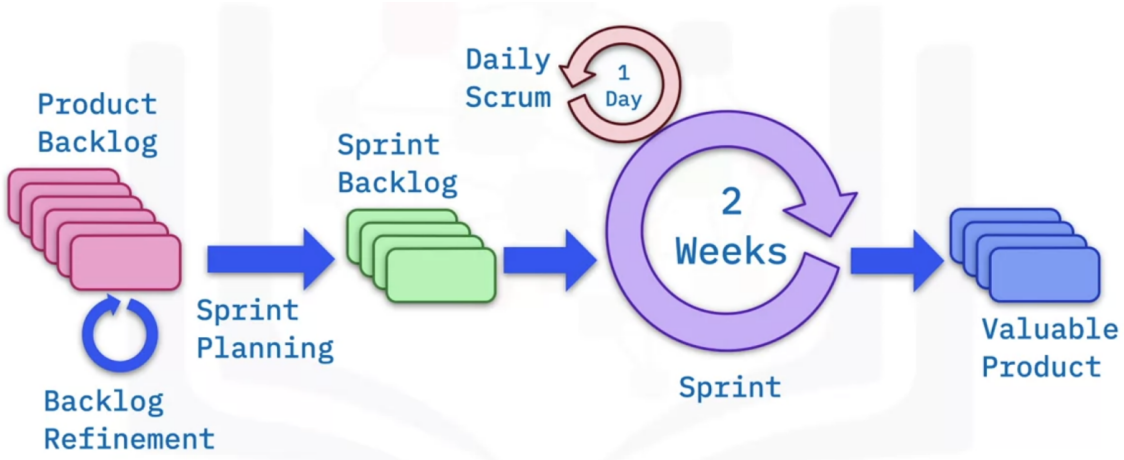
Scrum Overview

- Agile and scrum are not the same.
- Agile is a philosophy of doing work.
- Scrum is a methodology for doing work in an agile manner.
- **What is scrum?**
    - It is a management framework for incremental product development.
    - It emphasizes working in small, cross-functional teams.
    - Provides a structure of rules, roles, and artifacts.
    - Uses fixed-length iterations called sprints.
    - Has a goal to build a potentially shippable product increment with every iteration.

Sprint

- A sprint is one iteration through the design, code, test, and deployment cycle.
- Every sprint should have a goal.
- Sprints are usually 2 weeks in duration.

Steps in the Scrum Process



Rashid Wassan

Roles in Scrum

- **Product Owner**
  - Represents the stakeholder interests.
  - Articulate product vision.
  - Is the final arbiter of requirements questions.
  - Constantly re-prioritizes the product backlog, adjusting any expectations.
  - Accepts or rejects each product increment.
  - Decides whether to ship.
  - Decides whether to continue development.
- **Scrum Master**
  - Facilitates the Scrum process.
  - Coaches the team.
  - Creates an environment to allow the team to be self-organizing.
  - Shields the team from external interference to keep it in the zone.
  - Help resolve impediments.
  - Enforces sprint timeboxes.
  - Captures empirical data to adjust forecasts.
  - Has no management authority over the team.
- **Scrum Team**
  - A cross-functional team consisting of developers, testers, business analysts, domain experts, and others.
  - Self-organizing team with no externally assigned roles.
  - Self-managing as they self-assign their work.
  - Usually consists of 5-9 collaborative members.
  - Co-located.
  - Dedicated.
  - Negotiate commitments with the product owner.

Planning to be Agile

- **Destination Unknown:**
  - Plan iteratively.
  - Don’t decide everything at the point when you know the least.
  - Plan for what you know.
  - Adjust as you know more.
  - Your estimates will be more accurate.
- **Agile Roles and Need for Training**
  - Formulas of failure: assigning people new roles without training them.
  - Product Manager vs Product Owner
  - Project Manager vs Scrum Master
  - Development Team vs Scrum Team
- **Kanban and Agile Planning Tools**

Rashid Wassan

User Stories:

- **Story Contents**
    - Brief description of need and business value.
    - Any assumptions or details?
    - The definition of “Done”.
    - Given some conditions.
    - Gherkin syntax is used.
    - **INVEST Story**
        - Independent
        - Negotiable
        - Valuable
        - Estimable
        - Small
        - Testable
    - **Epic**
        - A big idea
        - A user story that is bigger than a single sprint
        - Backlogs usually start as epics and become stories when defined
        - For sprint planning, Epics need to be broken into stories
- **Effectively Using Story Points**
    - The story point is an abstract measure of the overall effort.
    - Used to measure the difficulty of implementing a user story.
    - Story points acknowledge that humans are bad at estimating time-to-completion.
    - Story points are like T-Shirt sizes.
    - Most tools use the Fibonacci series.
    - Since story points are relative, you will have to agree on what a particular size is.
    - **Story Size**
        - A story should be small enough to be coded and tested in a single sprint iteration, ideally, just a few days.
        - Large stories should be broken down into smaller ones.
    - **Story point anti-pattern**
        - Evaluating a story to wall-clock time.
        - Humans are bad at estimating wall clock time.

- **Building the Product Backlog**
    - The product backlog contains all the unimplemented stories not in a sprint.
    - Stories are ranked in order of importance or business value.
    - Series are more detailed at the top, and less detailed at the bottom.
    - **Sample requirements**
        - **As a …**
        - **I need …**
        - **So that …**
- **Lesson Summary - User Stories:**

● A user story documents a person requesting a function to achieve a goal.

- Using a template helps ensure that stories are complete.
- Defining "done" helps minimize misunderstandings.
- Use the INVEST acronym to remember the qualities of a good user story: independent, negotiable, valuable, estimable, small, and testable.
- Epics can be used to capture big ideas.
- Story points are a metric used to estimate the difficulty of implementing a given user story.
- Story points are relative, like T-Shirt sizes.
- You must agree on what "average" means.
- You should never equate story points with wall clock time.
- A product backlog is a ranked list of all unimplemented stories.
- Stories high in the ranking should have more detail than those that are lower.
- Create stories using the "As a", "I need", and "So that" template to ensure everyone understands who it benefits and the business value it provides.

Backlog Refinement

- Keep important stories on top.
- Break large stories near the top into smaller ones.
- Make sure that stories near the top of the backlog are groomed and complete.
- **Backlog refinement meeting:**
    - Who should attend?
        - Product owner.
        - Scrum master.
        - Development team (optional and only one of them would be enough)
    - What is the goal?
        - Groom the backlog by ranking the stories in order of importance.
        - Make sure the story contains enough information for a developer to start working on it.
- **New Issue Triage**
    - Start with new issue triage.
    - Goal: At the end of backlog refinement, the New Issues column is empty.
    - Take stories from new issues and move them into product backlog or icebox if of less priority. You can also reject them.
- **Backlog refinement workflow**
    - The product owner sorts the product log in order of importance.
    - The team may provide estimates and other technical information.
    - Large vague items are split and clarified.
    - The goal is to make stories “sprint ready”.
- **Labels**
    - Labels in GitHub
    - The yellow label should be for the technical dept as well.
    - **Examples of Technical Debt:**
        - Code refactoring.

        - Set up and maintain environments.
        - Changing technology, like databases.
        - Updating vulnerable libraries.
    - **Backlog refinement tips**
        - Refine backlog every sprint to ensure the priorities are correct.
        - Have at least two sprints' worth of stories groomed.

Sprint Planning

- Used to define what can be delivered in the sprint and how that work will be achieved.
- This is accomplished by producing a sprint backlog.
- **Sprint Planning Meeting**
    - Should be attended by Product Owner, Scrum Master, and the Development Team.
- **Goals:** Sprint goals should be clear. The product owner describes the goal and and product backlog items supporting.
- **Mechanics of Sprint Planning**
    - **The development team** takes stories from the product backlog and assigns them to the sprint backlog. It also assigns story points and labels. It also ensures each story has enough information for a developer to start working on it. It stops adding stories when team velocity is reached.
    - **Team Velocity** is the number of story points a team can complete in a single sprint. It can be changed over a period and it is unique to a team and cannot be compared to others.
- **Create a Sprint Milestone**
    - Create a sprint milestone to start the sprint.
    - The milestone title should be short.
    - The description should document the milestone goal.
    - Duration should be 2 weeks.

Lesson Summary: The Planning Process

- It is the product owner's responsibility to maintain a groomed backlog
- Backlog refinement is used to order the product backlog and make stories sprint ready
- You start refinement by triaging new issues
- Large stories should be broken down until they are small enough to fit in a sprint
- The goal of backlog refinement is to get the backlog ready for the sprint planning meeting
- It is the product owner's responsibility to present the sprint goal
- It is the development team's responsibility to create a sprint plan
- A sprint plan is created by moving stories from the product backlog into the sprint backlog until the team's velocity is reached

Daily Workflow:

- **Daily Execution:**

- Take the next highest priority item from the sprint backlog.
- Assign it to yourself.
- Move it in progress/process.
- No one should have more than one story assigned to them unless they are blocked.
- When you are finished, you create a pull request and move the story to QA.
- When the PR is merged, move the story to the ‘done’ column.

Daily Stand-up

- Occurs every day at the same time and place.
- AKA Daily Scrum.
- Each member briefly reports on their work.
- Timeboxed to 15 minutes.
- No one is allowed to sit.
- This is not a project status meeting.
- **Who should attend?**
  - Scrum master.
  - Development team.
  - Product onwer (optional).
- **Daily stand-up questions:**
  - What did I accomplish the previous day?
  - What will I work on today?
  - What blockers or impediments are in my way?
- **Impediments and blockers**
  - Impediments identified by the team should be unblocked by the scrum master.
  - Developers that are blocked should work on the next story.
- **Tabled topics**
  - Topics that are raised during the daily stand-up should be held until the meeting has ended.
  - Anyone interested in those topics can stay to discuss.

Lesson Summary: Executing the Plan:

You need to keep the Kanban board updated so that everyone knows what you are working on

- It is important to always work on the story with the highest priority that you have skills for
- Working on more than one story at a time may lead to neither story being finished at the end of the sprint
- The daily stand-up occurs every day for 15 minutes
- Topics not related to the stand-up should be addressed after the meeting
  Each person should be prepared to answer the three stand-up questions:
- What did I accomplish the previous day?
  What will I work on today?
  What blockers or impediments are in my way?

Rashid Wassan

Completing the Sprint

- **Milestones and Burndowns**
    - Milestones can be created for anything in your project.
    - Burndown charts are used to measure progress against any milestone.
- **Burndown Charts**
    - The measure of story points completed vs story points remaining for a sprint.
    - Over time remaining story points should go down, hence the name: burndown.

The Sprint Review

- It is a demo time.
- Live demonstration of implemented stories.
- The product owner evaluates if stories are done right.
- Done stories are closed.
- **Who should attend?**
    - Anybody can join (including optional customers).
- Feedback gets converted into new product backlog stories.
- This is where iterative development allows the creation of products that couldn't have been specified up-front in a plan-driven approach.
- **Rejected Stories**
    - What about stories that are not done?
    - Add a label to indicate this and close them.
    - Write a new story with new acceptance criteria.
    - This will keep the velocity more accurate since the work done on incomplete stories will also be counted.

The Sprint Retrospective

- A meeting to reflect on the sprint.
- Measures the health of the process.
- The development team must be comfortable speaking freely.
- **Who should attend The Sprint Retrospective Meeting?**
    - Scrum master.
    - Development team.
- **Three questions are asked:**
    - What went well?
    - What did not go well?
    - What should be changed for the next sprint?
- **The goal is the improvement**

Lesson Summary: Completing the Sprint

- A burndown chart shows the measurement of story points completed vs story points remaining for a sprint
- Burndown charts can be used to show progress and forecast the team's probability of achieving the sprint goal

- A sprint review is a demonstration of the features that have been implemented during the sprint
- Feedback from stakeholders is critical to help shape the future of the product
- The backlog is updated based on feedback
- A sprint retrospective is a time to reflect on how the sprint went
- The sprint retrospective is attended by the scrum master and the development team
- The team must feel comfortable speaking freely
- A sprint retrospective must result in changes to improve the next sprint
- Three questions are answered on what went right or wrong:

- What went well? (keep doing)
- What did not go well? (stop doing)
- What should we change for the next sprint?

Measuring Success

- Vanity metrics
- Actionable metrics

Getting Ready for Next Sprint

- End of sprint activities:
    - Move stories from done to close.
    - Close the current milestone.
    - Create a new sprint milestone.
    - Adjust unfinished work.
- **Handling Untouched Stories**
    - These stories can be moved to the top of the backlog.
    - Resist the urge to move them into the next sprint.
- **Handling Unfinished Stories**
    - Do not move unfinished stories into your next sprint.
    - Give the developer credit for the work they did.
    - This will keep the velocity more accurate.
    - Adjust the description and add an unfinished label and move it to the done.
    - Write a new story for the remaining work.
    - Assign the remaining story points and move them to the next sprint.
- **Ready for the Next Sprint**
    - All stories assigned to the current sprint are closed.
    - All unfinished stories are reassigned.
    - The sprint milestone is closed.
    - A new sprint milestone is created.

Agile Anti-Patterns and Health Check

- No real product owner.
- Multiple owners.
- Teams are too large ideal team should be less than 10.
- Teams are not dedicated.

- Teams are too geographically distributed.
- Teams are siloed.
- Teams are not self-managing.
- **Scrum Health Check**
    - Proper accountability.
    - Work is organized in consecutive sprints of 2-4 weeks or fewer.
    - There is an ordered product backlog.
    - Sprint backlog with visualization of remaining work for the sprint.
    - At sprint planning, a forecast, a sprint backlog, and a sprint goal should be created.
    - The result of the daily scrum is work being re-planned for the next day.
    - Stakeholders provide feedback from inspecting the increment at the sprint review.
    - The product log is updated as a result of the sprint review.