# Organizing Work Through Agile

From Themes to stories

# Roadmap

# What a roadmap is (not)

A roadmap can be many things. More important than what it is, is what it is not.

- A roadmap is not a release plan.
- It is not a listing of product features.
  - C. Todd example on Wodify, who posted their roadmap publicly:
  - They listed all of their features and the dates they would be released
  - Quickly started having trouble meeting those commitments.
  - After many missed dates and slipped projects, they removed the dates from their roadmap.

# What a roadmap is

A roadmap is about:

- Communicating product strategy rather than features and dates.
- It is a statement of intent and direction
  - like a literal road map for traveling, it states a destination headed towards and provides options of a path to get there.
- A roadmap tells us how we will realize our product vision.
- A roadmap focus on the outcomes to achieve.
- A release plan tells the outputs that will be ship along the way.

# TOP 5 components of a roadmap

**Product vision**

Business objectives

Timeframes

Objectives / Themes / initiatives

Disclaimer

**How a future world will benefit** from your product when it is fully realized. A product vision should align with the company vision.

Google To organize all of the data in the world and make it accessible for everyone in a useful way.

To provide a fast, simple, and secure browser for everyone to experience the modern web.

▶ To give everyone a voice and show them the world.

Helps people around the globe enjoy great access to information and opportunity than ever before.

# TOP 5 components of a roadmap

- Product vision
- **Business objectives**
- Timeframes
- Objectives / Themes / initiatives
- Disclaimer

## Defined Objectives to accomplish

# TOP 5 components of a roadmap

- Product vision
- Business objectives
- **Timeframes**
- Objectives / Themes / initiatives
- Disclaimer

**Timeframes should allude to prioritization**, give sequencing and guidance on timing. Showing a clear priority and broad timeframes helps frame the conversation.

# TOP 5 components of a roadmap

Product vision

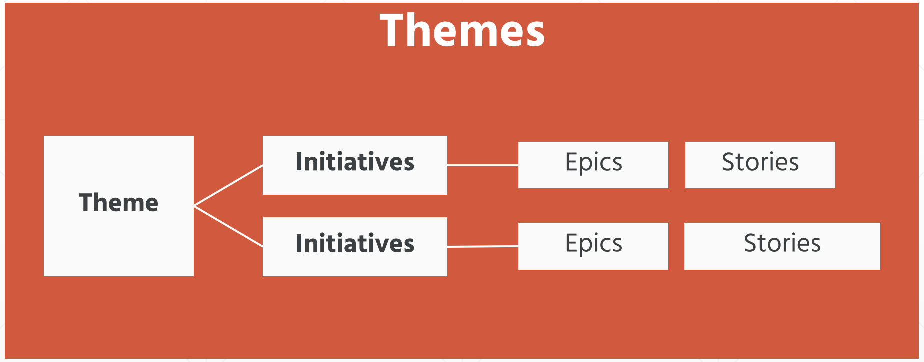
Business objectives

Timeframes

Objectives / Themes / initiatives

Disclaimer

**Themes are an organization of work-areas for our product**, outlining the customer or business outcome you are trying to achieve to fix a problem raised by the customers.



# TOP 5 components of a roadmap

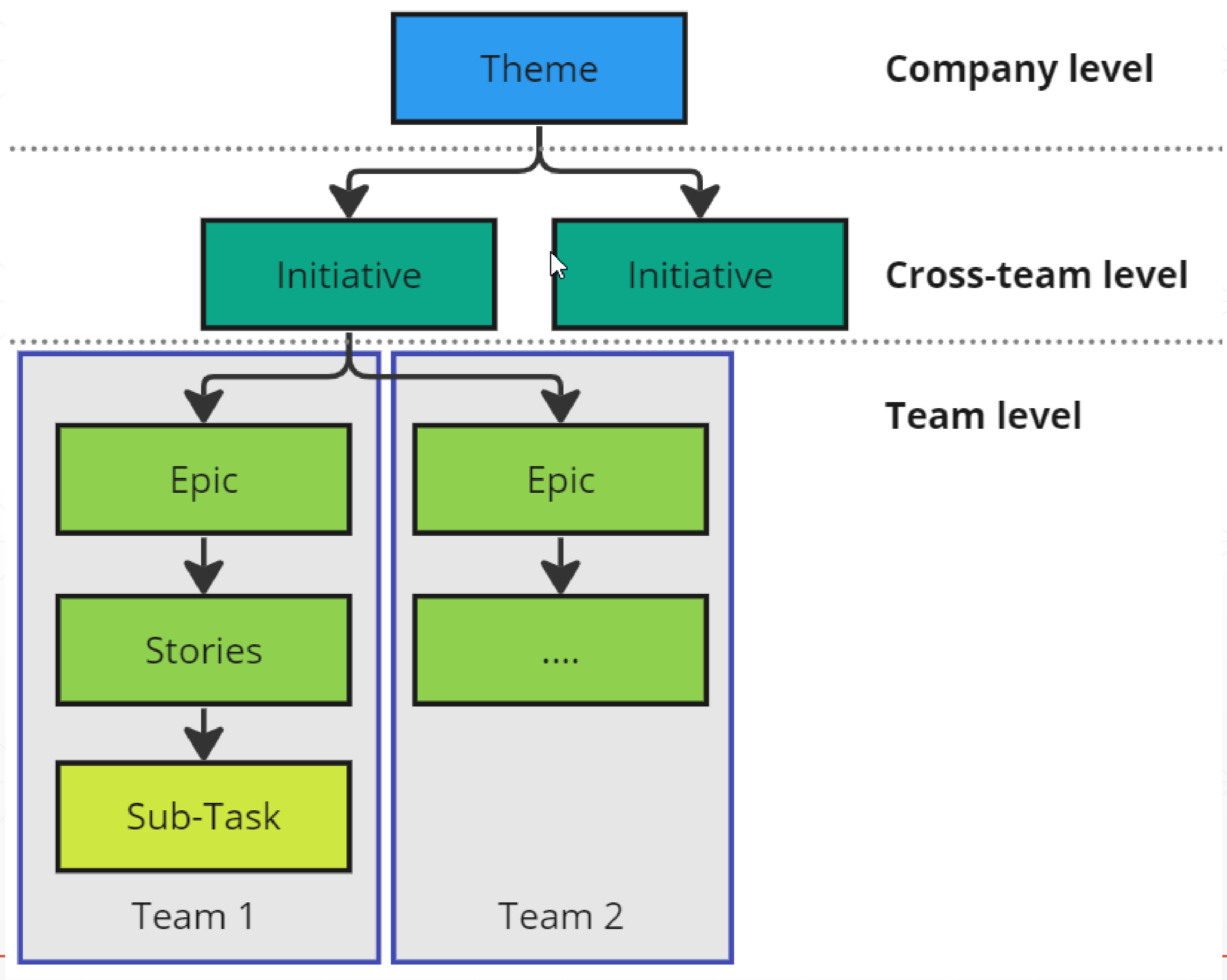
- Product vision
- Business objectives
- Timeframes
- Objectives / Themes / initiatives
- Disclaimer

**A disclaimer detailing the constraints and the likelihood of change** is incredibly important. It protects you from accusations of broken promises, and makes it clear that change is possible, even likely.

# From Themes to Stories

From Themes to stories

# From Theme to Stories



# Themes

# What are a Theme?

- Themes track the high level goals the organization
- Very high-level goals

- Spotify example: “becoming the leading streaming platform for listeners”

# Initiatives

# What are Initiatives?

- Initiatives are drivers of Epics
- There might multiple Epics running simultaneously under an initiative
- Can take 1 year
- Can be done by different teams

# Epics

# What are Epics?

- Collections of stories that are working towards a particular goal
- Represents a Minimum Viable Product (MVP)
- Usually from a month to a quarter

# Writing an Epic

- Describe the context
- Explain the value that it will bring to customers and how it supports the product strategy and business growth
- Explain the overall solution (specifications of the design phase)
- Have the success metrics and defined ways to measure it
- Link all relevant documents (story mapping, high-fidelity designs, and others)

# User Stories

# Stories

- Stories are the smallest component of agile framework
- They’re written from the point of view of the user
- Keep the team working user-focused
- They use the selected user stories of the specifications of design phase
- They increment customer value

# What are User Stories?

- Description of a feature through the eyes of a user
- Briefly encapsulate user persona, the job they need to accomplish, and what their overall needs are
- A user story doesn’t state what the team should build in order to fill the need
- User stories put the voice of the customer at the centre and keep every member of the team focused on their needs
- Can be completed every week

# User Stories Benefits

- Keep the teams’ focus on the user, and how a certain product feature will benefit them.
- Help teams to manage workflow with both speed and flexibility.
- Encourage collaborative, creative thinking by stating the why but not the how
- Empower designers and engineers to build with the right goal in mind.

# How user stories fit into agile frameworks?

- Scrum: They are the **product backlog** and are prioritized collaboratively by **scrum teams**
- Kanban: They are the **product backlog** and are prioritized collaboratively by **the Product Owner**

# Writing an User Story

**As a** <user role> **I want to** <action/behaviour> **so that** <benefit>

# Refining User Stories

# User Stories

Explanation

# Refining a user story

- As a <user role> I want to <action/behaviour> so that <benefit>

"As a [persona]": Who are we building this for?

“Wants to”: Here we’re describing their intent — not the features they use. What is it they’re actually trying to achieve?

“So that”: how does their immediate desire to do something this fit into their bigger picture? What’s the overall benefit they’re trying to achieve? What is the big problem that needs solving?

# Refining a user story

*Example:*
*As a online book buyer I want to buy a new book paying by credit card so that I can order the book.*

We identify the user (online book buyer), the action (buy), the data (new book and credit card).

Refining from it (examples):

Business rules: The credit card expiration date is on 10/10/2025

Interfaces: System interface with product inventory

Environment: Online shopping

Quality atributes: Transaction must take less than 10 seconds

Security atributes: The communication with the credit card provider must be secure

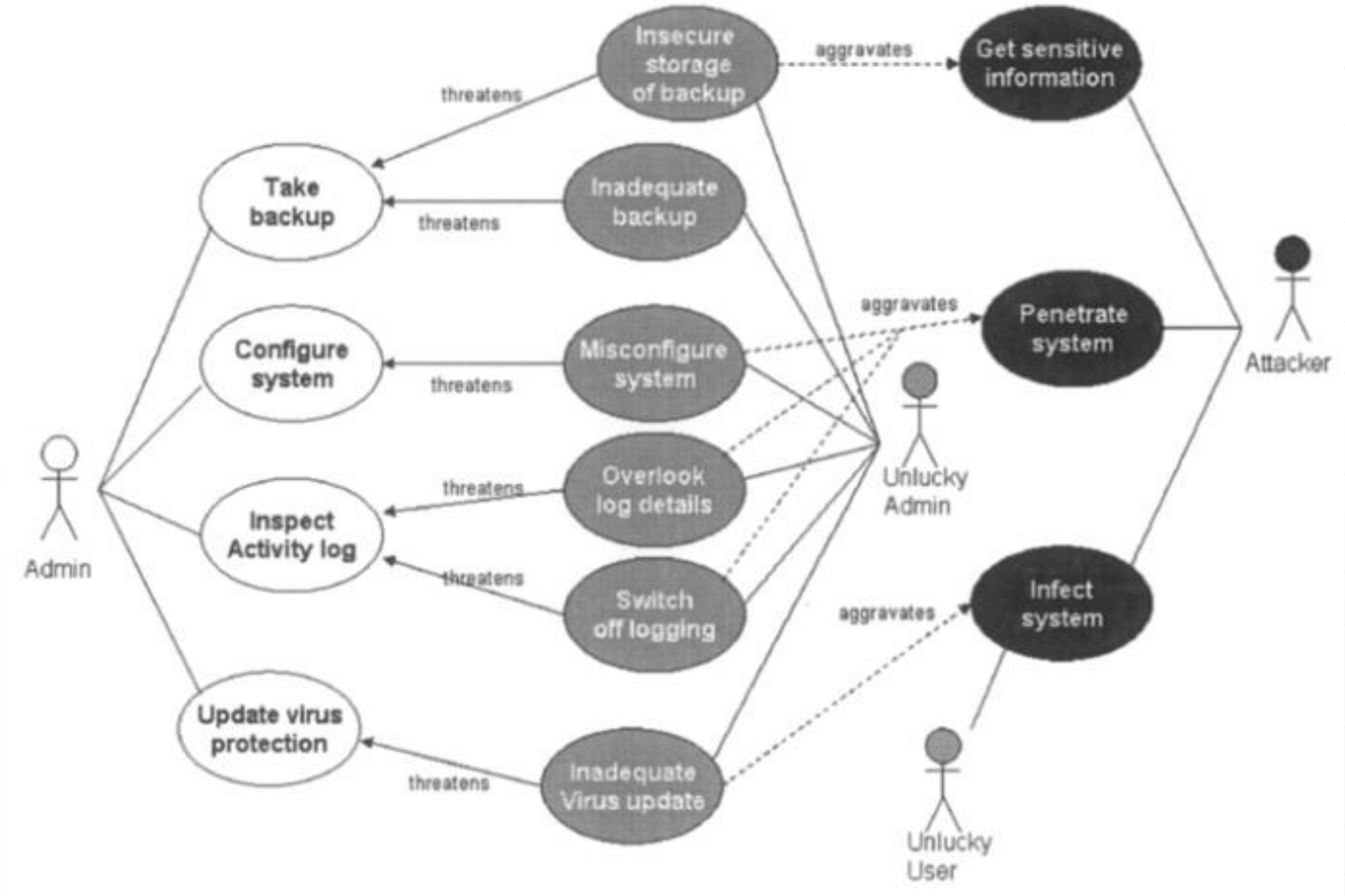
…

# Craft – Design – Acceptance criterias

Abuse cases and Misuse cases should also be described if needed taking in consideration the main user story

**Misuse cases** are use cases from a user that uses the system incorrectly potentially creating issues and anomolies

**Abuse cases** are usecases from user that try to exploit the system to take advantage of it



(Sindre, 2007).

# Craft – Design – Acceptance criterias

**Acceptance conditions** are the conditions that must be satisfied for the work to be accepted

- They can follow the scenario based-template (Gherkin):
  - Given (some given context or precondition),
  - when (I take this action),
  - Then (this will be the result)
    - And/but (optionals to give more restrictive behaviours)
- Checklist

Acceptance criterias should be:

- Testable
- Leave no room for bad interpretation
- Clear and concise
- Understandable for everyone
- Provide user perspective

# Refining a user story – Acceptance Criteria

**Given** - used to describe the initial context of the system - the scene of the scenario. It is typically something that happened in the past.

- The system is described to be in a well-defined state, such as creating and configuring objects or adding data to a test database.
- Their purpose is to put the system in a known state before the user (or external system) starts interacting with the system (in the When steps). Avoid talking about user interaction in Given.
- It’s okay to have several Given steps (use And or But for number 2 and upwards to make it more readable).

# Refining a user story – Acceptance Criteria

**When** - When steps are used to describe an event or an action. This can be a person interacting with the system, or it can be an event triggered by another system.

- It’s strongly recommended you only have a single When step per Scenario. If you feel compelled to add more, it’s usually a sign that you should split the scenario up into multiple scenarios.

# Refining a user story – Acceptance Criteria

**Then** - Then steps are used to describe an expected outcome, or result.

- An outcome should be on an observable output. That is, something that comes out of the system (report, user interface, message), and not a behavior deeply buried inside the system (like a record in a database).
- You should only verify an outcome that is observable for the user (or external system), and changes to a database are usually not.
- And, But - If you have successive Given’s, When’s, or Then’s, you can use And and But to add new lines to the requirement.

# User Stories

Exercise