WEB OF
THINGS

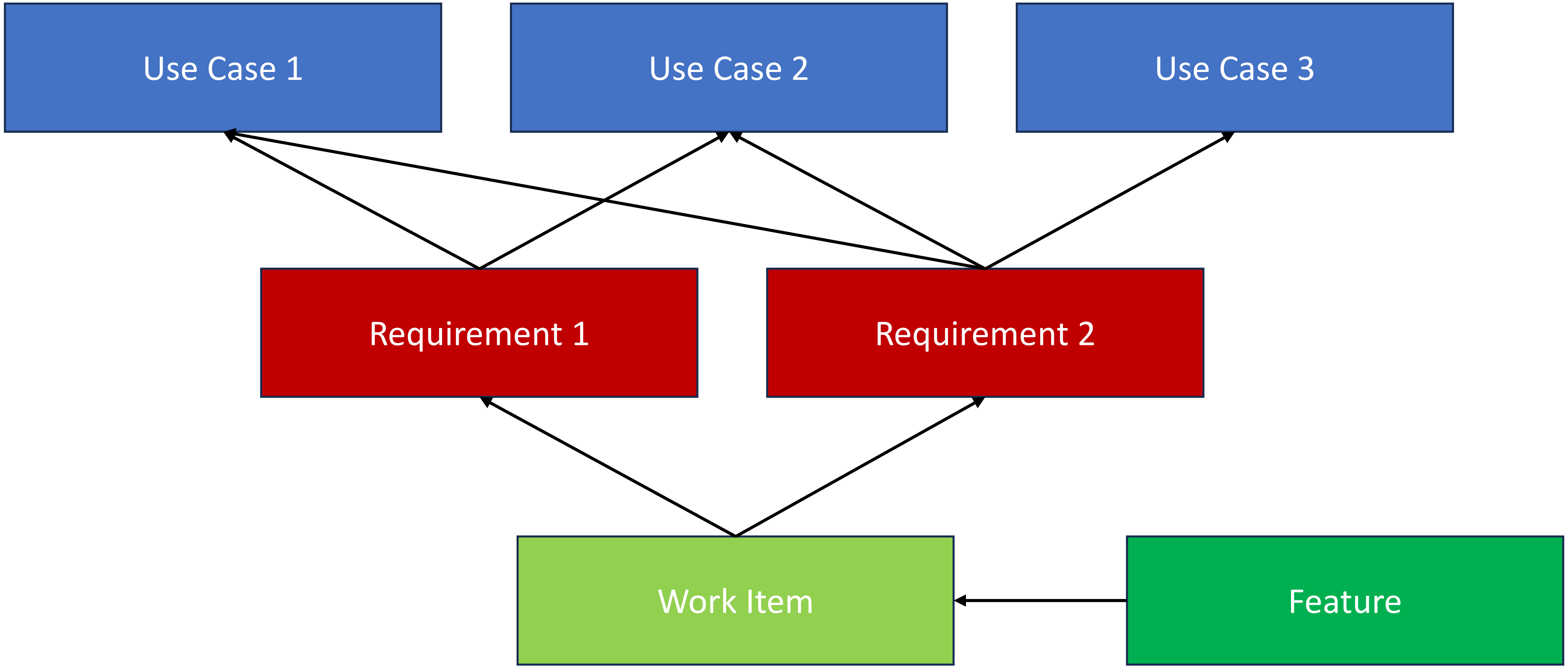
Requirements

Michael McCool
26 September 2024
TPAC 2024

Outline

- Purpose
  - Connecting use cases to features
- Format
  - User stories
  - Function vs technical
  - Examples
- Categories
- Security and Privacy
  - Relationship of requirements to risks
- Suggested Plan

UC → Req → WI → Feature



Requirements: Format

- User Story Template:

  As a **PERSONA**, I want **GOAL** so that **PURPOSE**.

PERSONA:

- Identifies primary user or beneficiary.
- Secondary stakeholders may also be identified in **PURPOSE**.

GOAL:

- May or may not map onto a single specific work item or feature (discuss).
- Should be specific enough that it is clear when it is satisfied.
- Should be satisfiable with finite effort.

PURPOSE:

- Larger context of goal, other stakeholders (e.g. support another SDO)
- May not be finitely satisfiable.

Functional vs. Technical Requirement

Functional:

- Requirement does not specify any particular features.

Technical:

- Requirement does specify a particular feature or set of features.

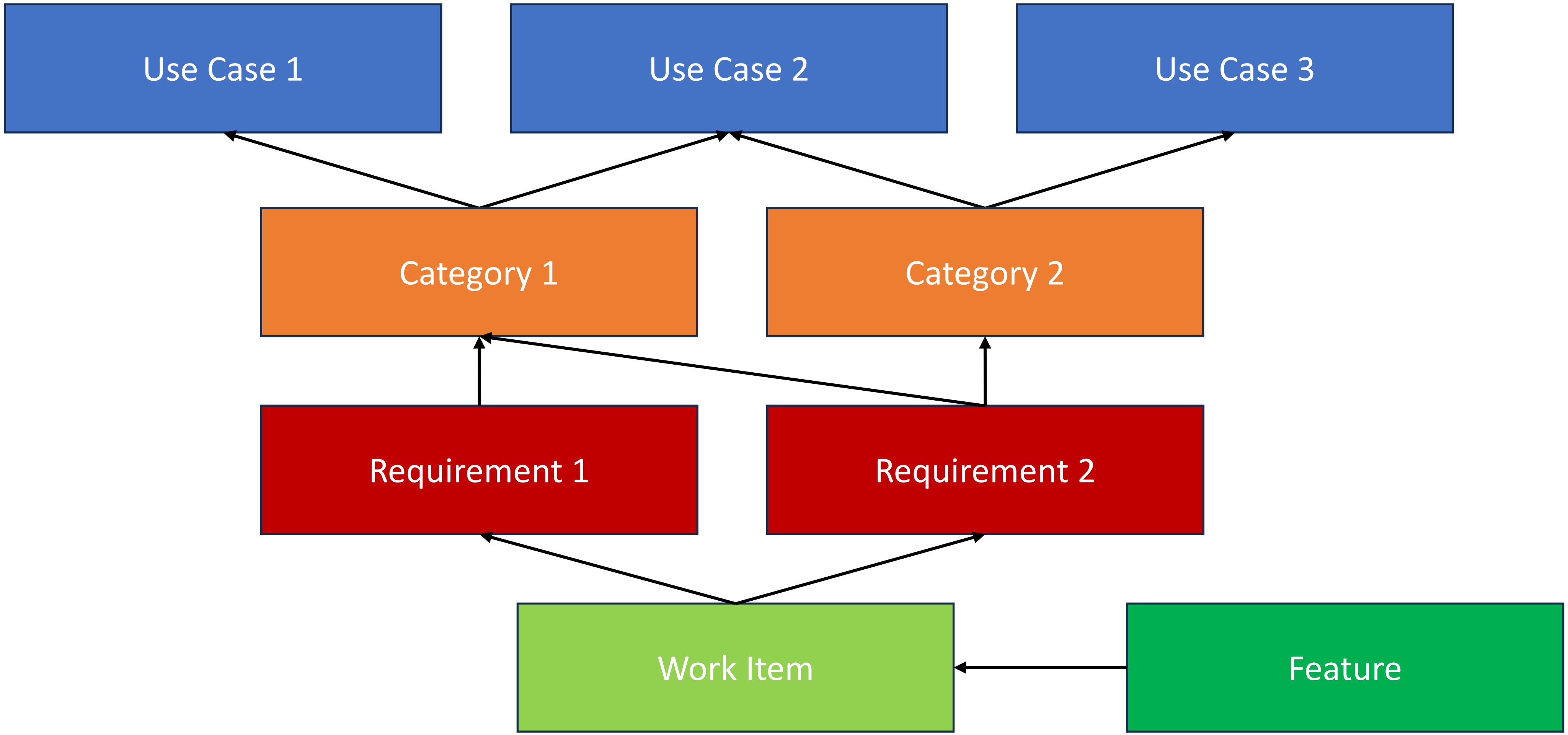
Proposal:

- Don’t worry too much about functional vs. technical…
- The main point is that requirements should be finitely satisfiable.
  - For example “WoT systems should have good security” is a bad requirement, it’s unclear when it is (or can ever be) fully satisfied.

Requirements: Examples

- As a ***consumer of WoT TDs***,
  I want ***the ability to poll the status of actions***
  so that ***I can take corrective action or cancel unneeded actions.***
- As a ***WoT System Owner***,
  I want ***to be able to control who has access to individual entries in a WoT TD Directory and revoke their access at any time***
  so that ***the security of my system can be maintained.***
- As a ***producer of WoT TDs,***
  I want ***to be able to publish a short form of TDs specifying only the variables to be filled into a TD Template***
  so that ***network bandwidth can be minimized.***

UC → Cat → Req → WI → Feature



Categories

- Intermediate step but allows for generalization
  - Avoids having to constantly update “requirement” to “use case” mapping
  - ***JUST A CONVENIENCE*** when many use cases share common requirements

Some possible categories:

- Private (handles personal or confidential information)
- Flexible Protocol Usage (systems use multiple protocols)
- Cloud Integration (needs to share data with remote servers)
- Local Access (needs to operate without a global connection)
- Mobile (location is subject to change)
- Resiliency (needs to be robust to failures and attacks of various kinds)

Security and Privacy Requirements

Special case:

- Security/Privacy features are generally to mitigate “risks”
  - S&P sections generally each have a defined risk
  - … then mitigations, some of which may be normative
- Risks are documented in “Security and Privacy Guidelines” document
  - Stakeholders need to be made consistent with other documents
- ***Need to identify which use cases have which risks***

Suggested Plan

- Expand “Requirements” Section in Use Cases and Requirements document to define requirements and connect them to use cases.
  - Want to avoid editing use cases themselves for authorship and consolidation reasons
  - Consolidate: move requirements out of other documents, e.g. Architecture
- Keep it simple:
  - A named requirement and a user story defining each one is enough.
  - Can link to another document for more detailed definitions, e.g. security risks.
    - However, best IMO if any liked details are in a “published” document, not a random MD file somewhere…
  - Links to use cases motivating each requirement
  - Do not have to link each requirement to ALL use cases motivating it
    - Use categories *only* if requirement motivated by large set of use cases
  - **Discuss:** do we want to allow “hierarchies” of requirements?
    - The PURPOSE does this, partially.

Discussion