



# Agile Requirements & Vision Hierarchy

Scalable Software Engineering  
WS 2021/22

Enterprise Platform and Integration Concepts

# Taxonomy of Goals & Tests



## Behavior-Driven-Development

- BDD provides multiple levels of testing
  - From coarse to fine
  - Large acceptance tests to small unit test

## Taxonomy of Goals

- **There are project goals larger than acceptance tests**
- Taxonomies provide common language



# Agile Vision Pyramid



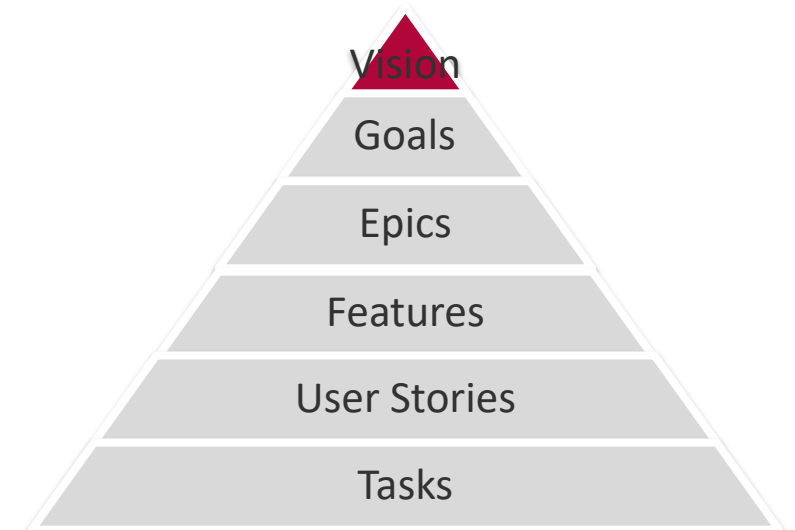
There is no set standard here!  
*Features aka Use Cases*  
*User Stories aka Scenarios*

## All Stakeholders, one statement

- Broad scope, could take multiple projects
- *Timeframe*: multiple years or decades
- *Example*: **Be the preferred banking provider for small business customers**

## Core stakeholders define the vision

- Incidental stakeholders help understand
  - What is possible
  - At what cost
  - With what likelihood

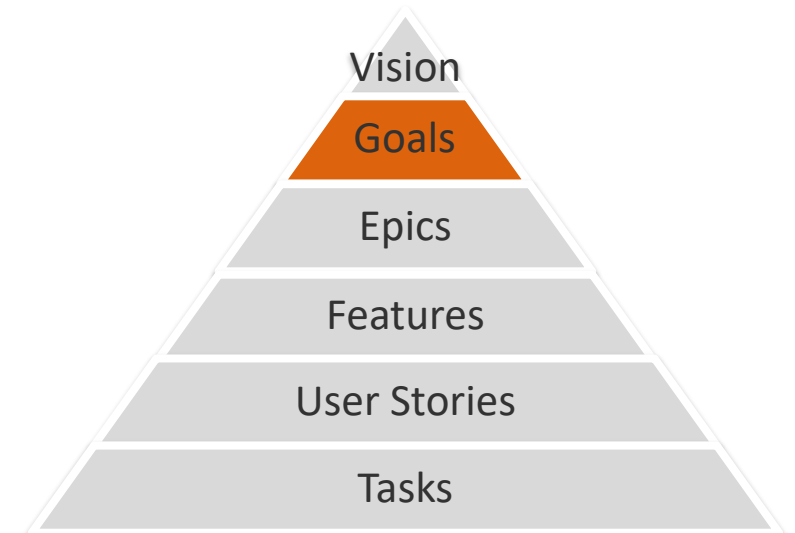


## How the vision will be achieved

- Aspects of the vision
- *Timeframe*: multiple months or years
- *Example*: **Increase engagement of small business customers with website**

## Can include measurements for success:

- Increase engagement with website by 20%
- Engagement measured by time spent on website vs. in support calls

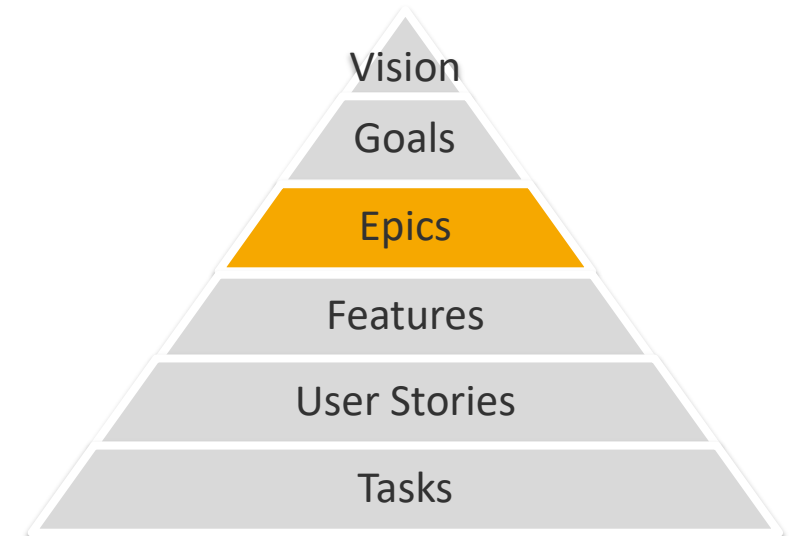


# Epics



## Large, cohesive blocks of of business value

- Too high level to start coding, useful for conversations
- May take several teams and sprints, but have defined end
- *Timeframe*: More than one iteration, possibly several releases
- *Example*: **Add a Web Self-Service Portal for common needs of small businesses**



# Features

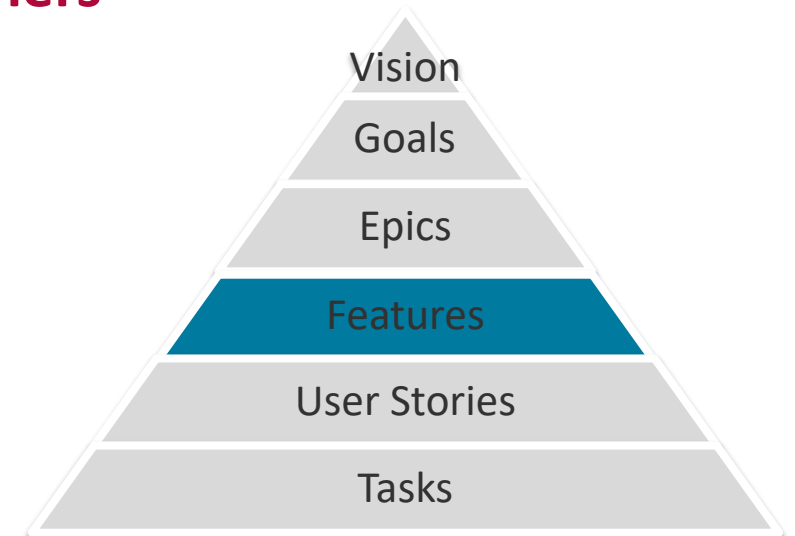


## Cohesive blocks of business value addressing a particular need

- Describe the behavior to be implemented in software
- Can be traced back to a stakeholder
- Release notes can be built based on the list of recently completed features
- *Timeframe*: within a single release or even in single iteration
- *Example*: **Manage the transactions with business partners in separate accounts**

## Warning:

- Starting directly at this level might miss lots of context



# User Stories



## Demonstrable functionality

- Incremental value the team delivers to create a feature
- A single completed story may not provide meaningful business value
- Vertical cut through technology stack (e.g. not database-only)
- Should conform to INVEST attributes
- *Timeframe*: Within an iteration
- *Example*:  
As the **sales specialist of a small company**  
I want to **create separate IBANs for individual customers**  
so that I can **structure payments and chargebacks**



See <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>



# User Story Format



**Informal, natural language narrative from perspective of end user**

## ■ Structure

- **Role, reason, benefit** (*why?*)
- “As a <role> I can <capability>, so that <receive benefit>”
- “As <who> <when> <where>, I want <what> because <why>”

## ■ Includes priority and effort estimate

## Acceptance criteria

- What is required to consider story as complete
  - Related to Definition of Done
  - E.g. in *Given-When-Then* format



# Tasks

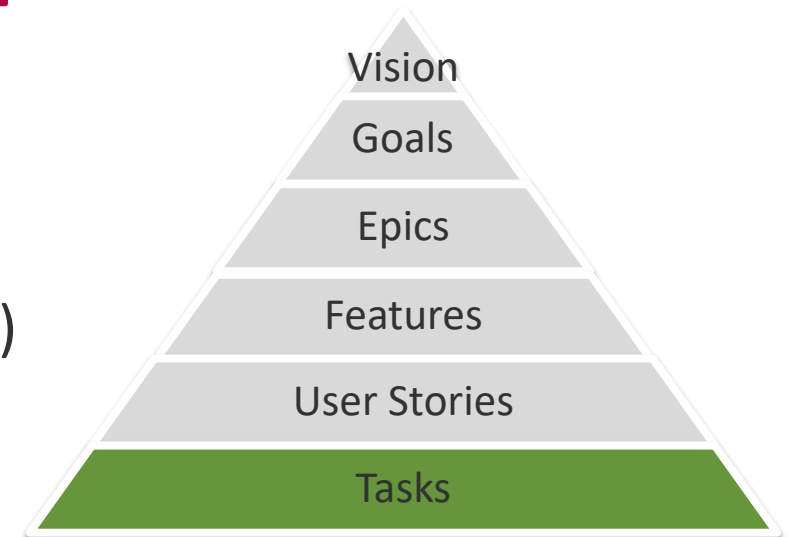


## The work required to complete a story

- Tasks are identified during sprint planning
- Usually defined by the people doing the work
- No need to be understandable by business users, can be highly technical
- *Timeframe*: hours, maybe a day
- *Example*: **HTML view with button to request new IBAN**

## Stories vs. Tasks

- Stories: multiple types of work  
(e.g., programming, testing, database design, UI design)
- Tasks: restricted to a single type of work



# Line of Autonomy

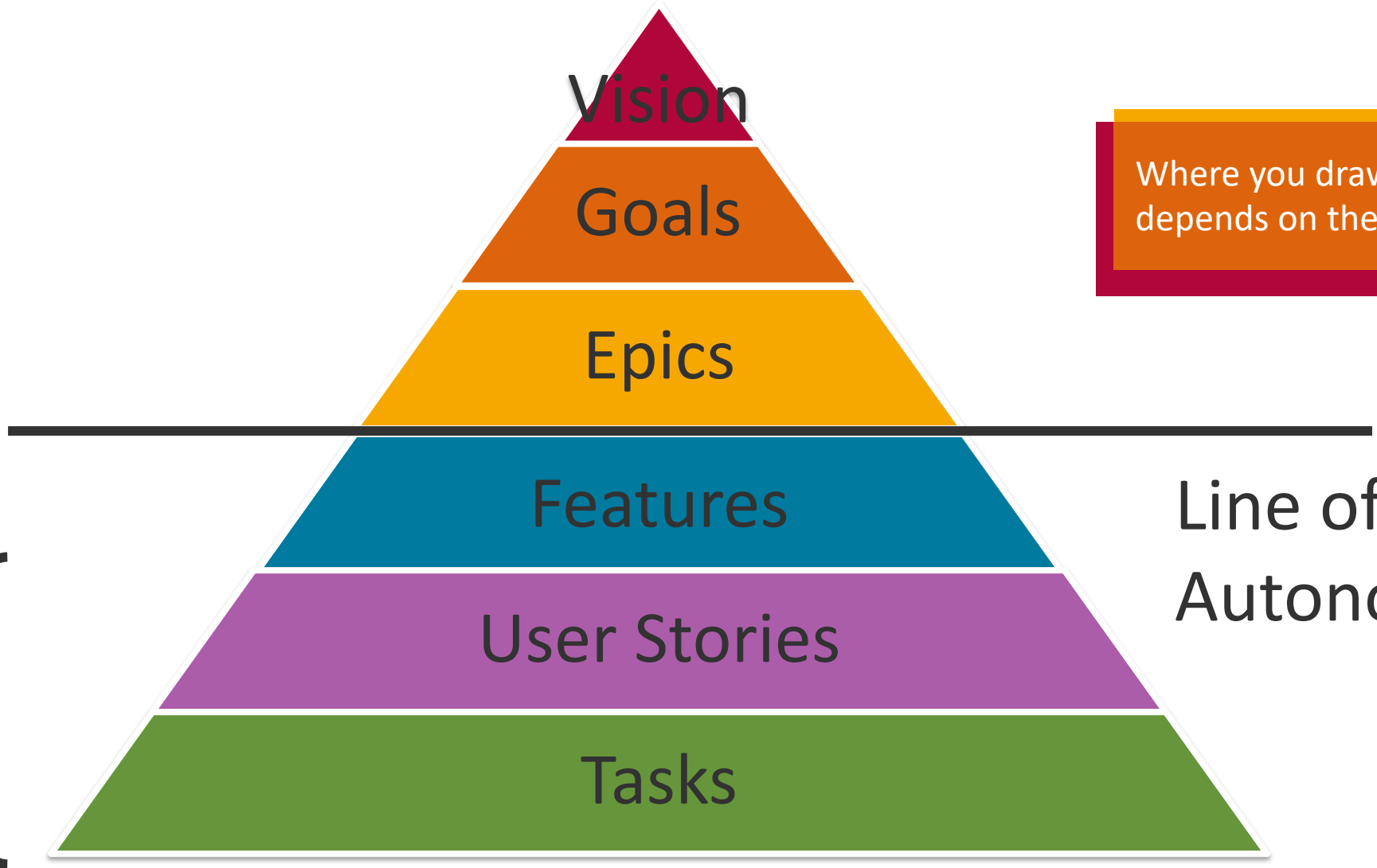


## Inflection point between strategic direction and tactical action

- Scrum Sprints meet longer-term planning
- Point in hierarchy where team is the clear owner; management doesn't interfere
- Project management: **Alignment** through higher level goals
- Teams: **Autonomy** by deciding how to work



# Line of Autonomy

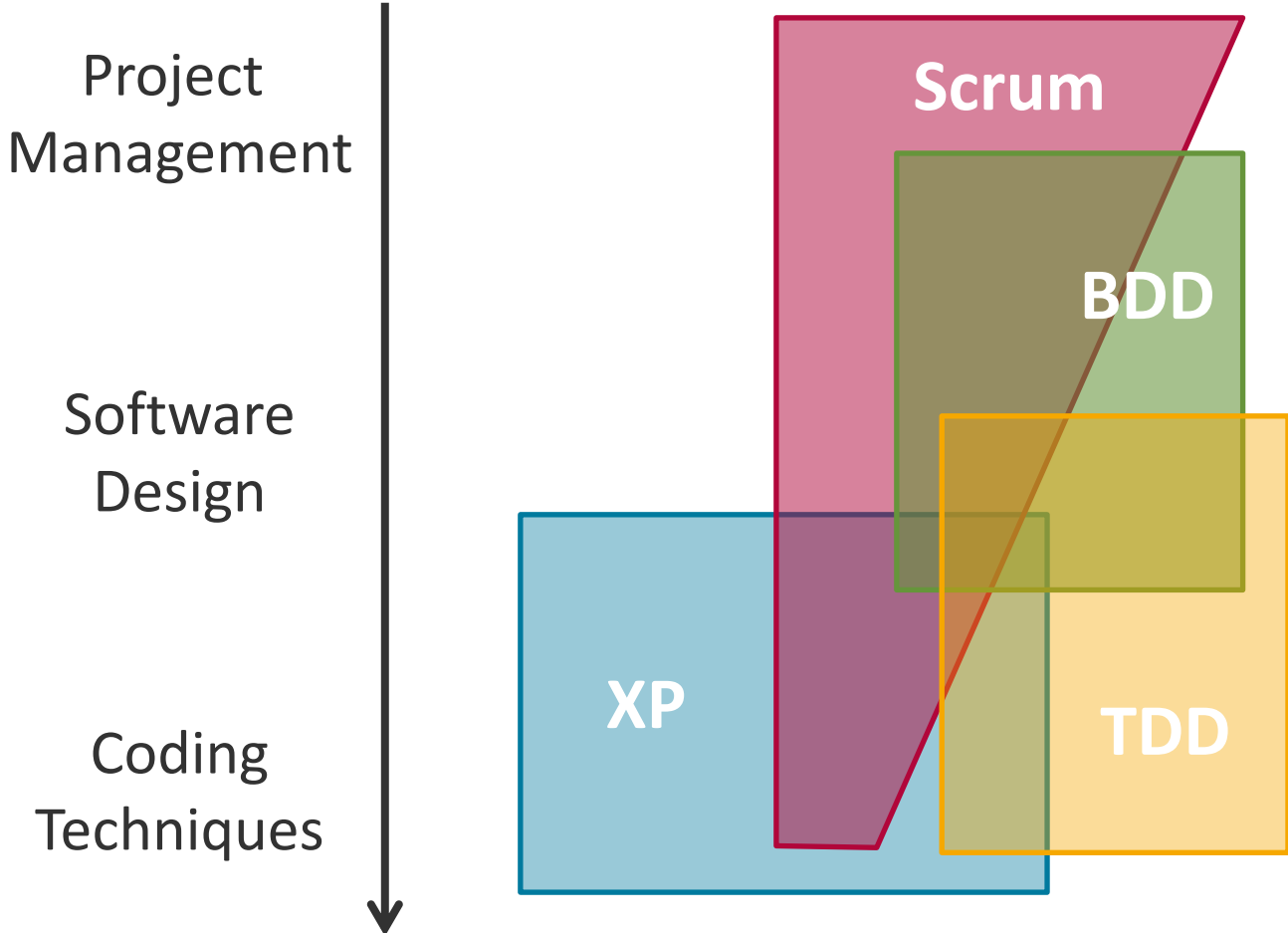


Where you draw the line also depends on the context

Scrum {

Line of  
Autonomy

# Agile Methods & BDD



# BDD and Testing



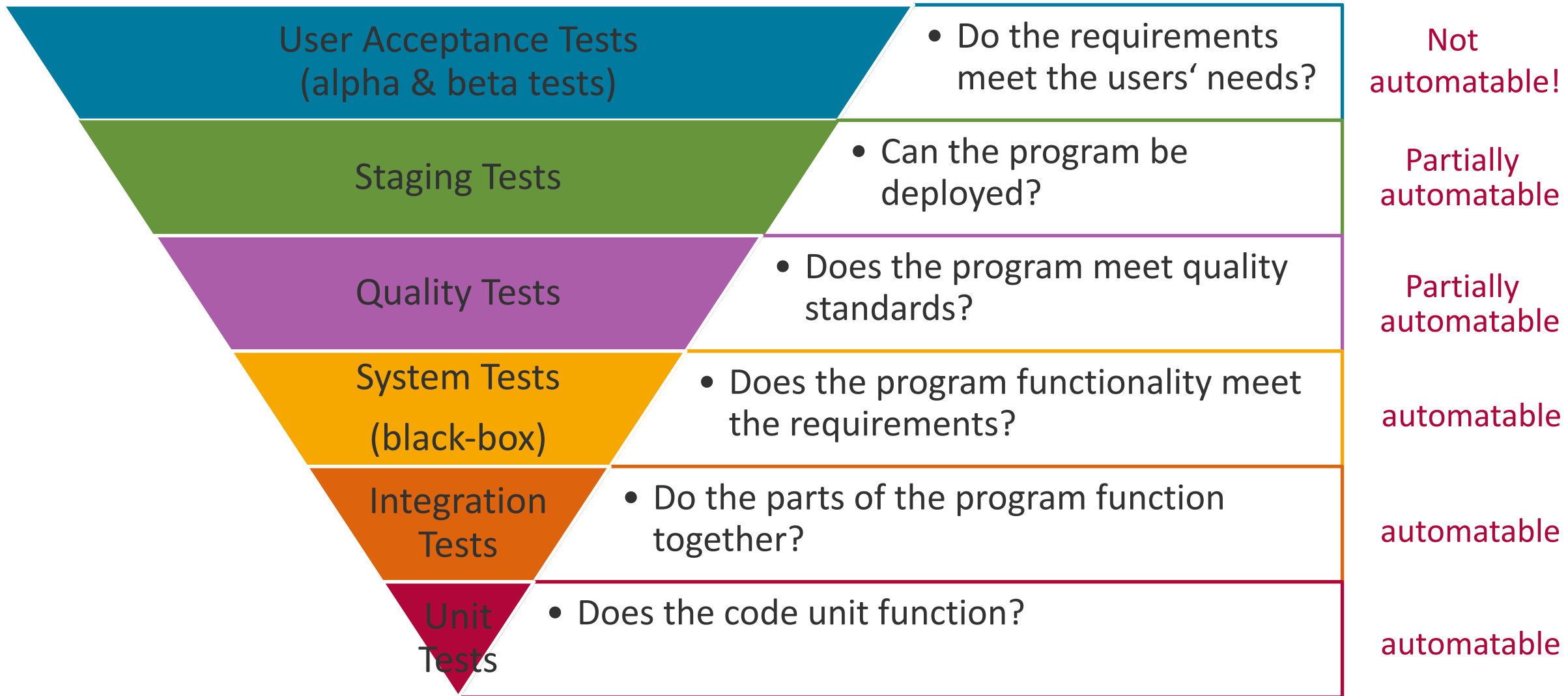
## For stakeholders

- Story-based definition of application behavior
- Definition of features to reach goal & vision
- Business value is specified in requirements

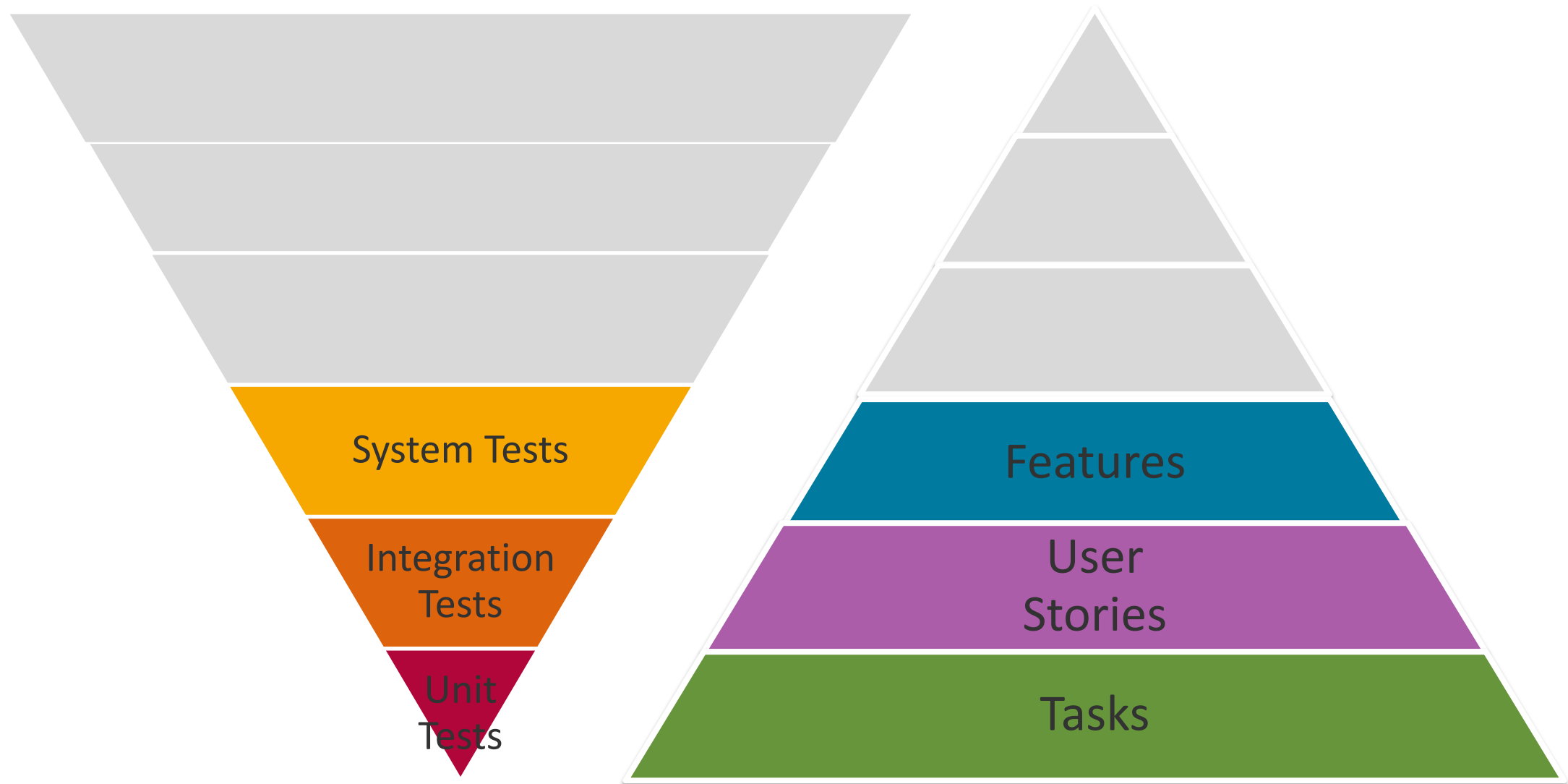
## For the developer

- BDD Cycle, definition of stories/tests with PO
- Coding with TDD/test-first approach

# Levels of Testing



# Hierarchy of Goals vs Test Levels





# Summary



## Taxonomy of Agile Goals

- Agile Requirements Pyramid
  - Vision -> user stories -> tasks
- Line of Autonomy
- Agile Methods & BDD
- Levels of testing vs. levels of goals

