

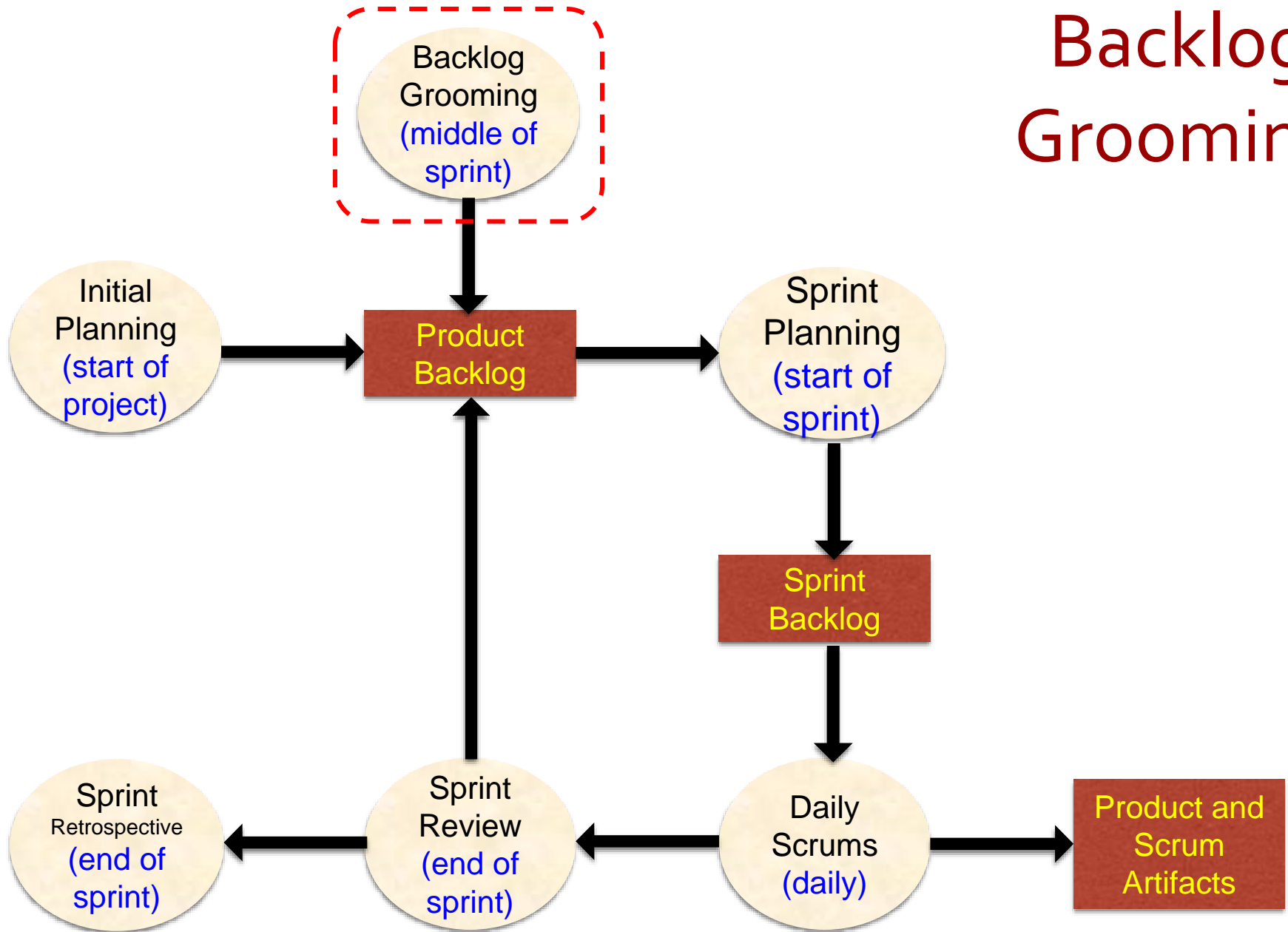
# Sprint Activities: Daily Scrum

- AKA “Stand-Up Meeting”
- Purpose: coordination and short-term planning
- Held daily (when Developers work full-time)

# Sprint Activities: Daily Scrum

- Three Questions for each team member:
  - What did you accomplish since yesterday?
  - What do you plan to do today?
  - What impediments block your progress?
- Impediments that can't be solved in the meeting become action items for the Scrum Master

# Backlog Grooming



# Sprint Activities: Backlog Grooming

- AKA Product Review, Backlog Review, and Sims & Johnson refer to it as Story Time
- Crucial and sometimes overlooked in practice!
- Held at least once/sprint or more often as necessary
- Purpose: update Product Backlog to reflect “current version of the truth”

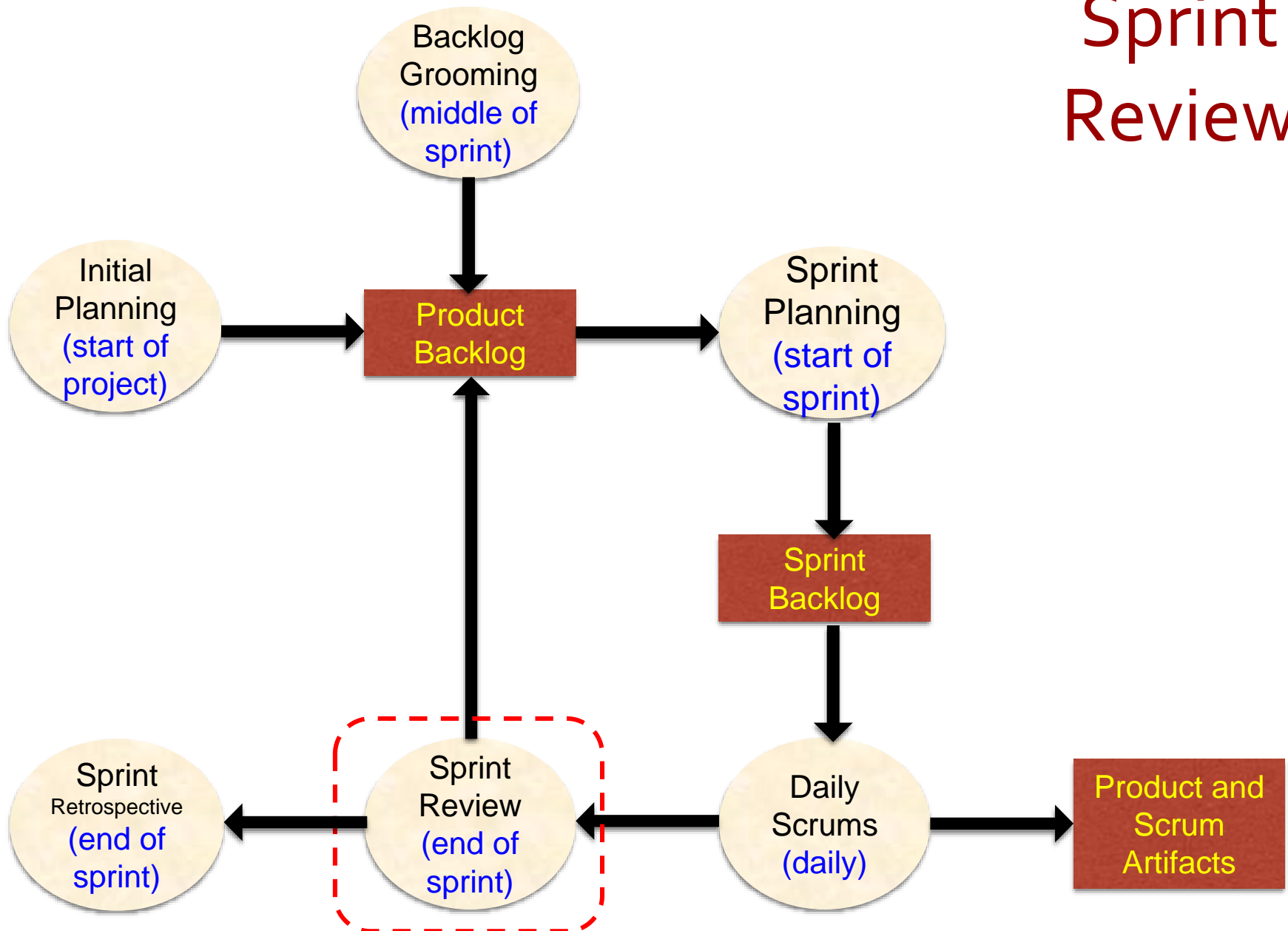
# Sprint Activities: Backlog Grooming

- Decompose epics (complex stories) into smaller stories
- Revise/remove stale stories
- Define/refine Acceptance Criteria for stories
- Product Owner (not Developers) may re-prioritize stories
- Developers revise/update stale estimates

# Sprint Activities: Backlog Grooming

- Decompose epics (complex stories) into smaller stories
- Revise/remove stale stories
- Define/refine Acceptance Criteria for stories
- Product Owner (not Developers) may re-prioritize stories
- Developers revise/update stale estimates
- Take-away message: Backlog Grooming meeting prepares the Backlog for the **next** sprint. It does not affect any work that is done in the current sprint.

# Sprint Review



# Sprint Activities: Sprint Review

- AKA Sprint Demo
- Held at end of sprint
- Purpose: Demonstrate completed User Stories to
  - stakeholders / client
  - entire team
- Ideally: Have customer use the product



# Sprint Activities: Sprint Review

- Stakeholders and entire team invited
- Requires about 1 hour per week of sprint
- Enumerate (not demonstrate) incomplete (snow plowed) work
- In CS471 teams will demo their product to sponsors

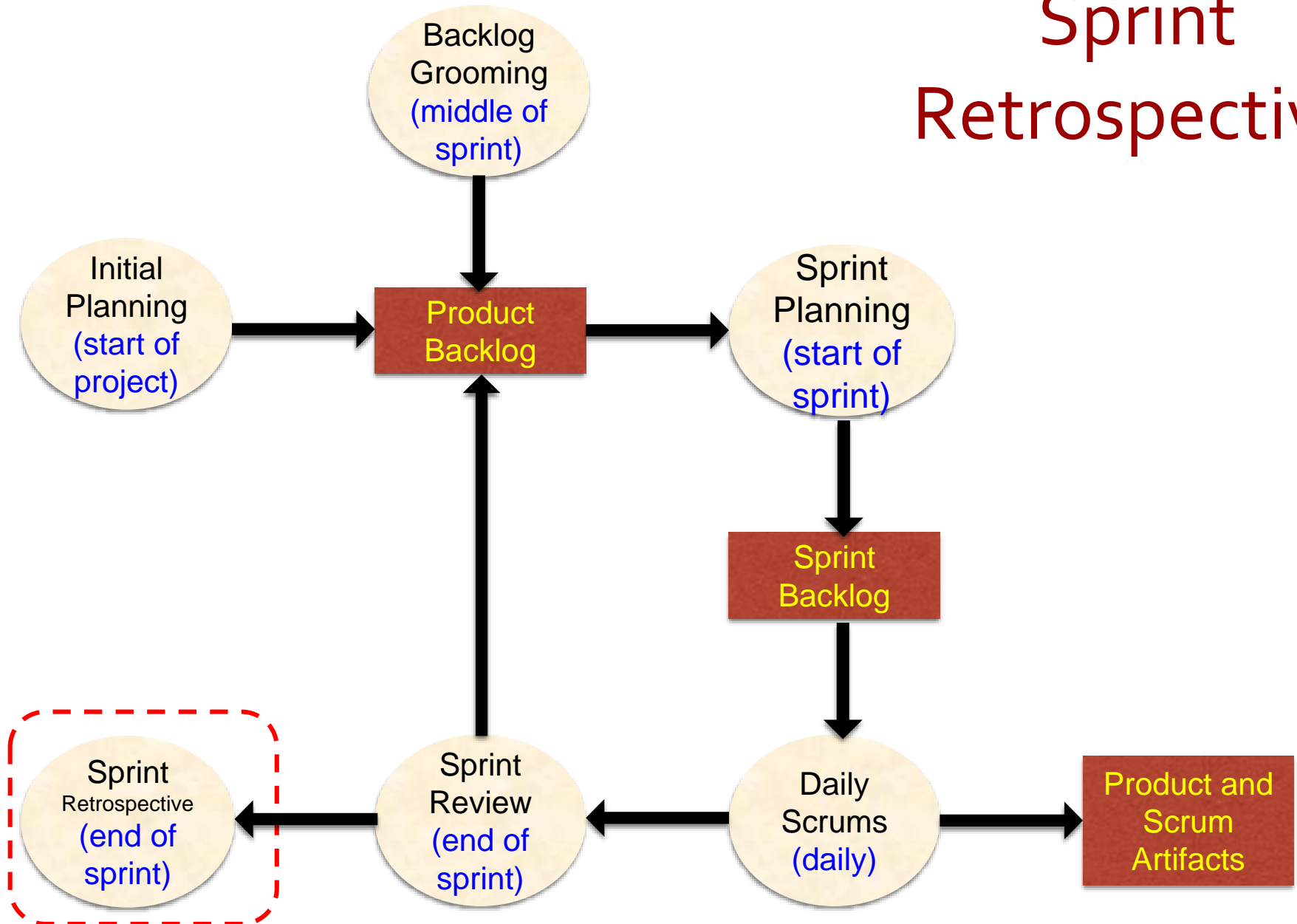
# Sprint Activities: Sprint Review

- Customer/user feedback is integrated back into the Product Backlog:
  - How?

# Sprint Activities: Sprint Review

- Customer/user feedback is integrated back into the Product Backlog:
  - Create new user stories/acceptance criteria
  - Refine existing user stories/acceptance criteria
  - Create bug reports

# Sprint Retrospective



# Sprint Activities: Sprint Retrospective

- Held after Sprint Review
- Only Development Team and Scrum Master attend
  - NB: Product Owner & Stakeholders DO NOT attend
- Purpose: Continuous development process improvement

# Sprint Activities: Sprint Retrospective

- Reflection on past sprint activities:
  - Choose one or two process improvements to implement in next sprint:

# Sprint Activities: Sprint Retrospective

- Reflection on past sprint activities:
  - Choose one or two process improvements to implement in next sprint:
    - use of pair programming
    - use of pull-requests
    - use of code-review
    - use/change of static analysis tools
    - use of a management tool over another, etc.

# Sprint Activities: Sprint Retrospective

- Team uses **Start/Stop/Continue** exercise on these **process improvements**. What **activity** we:
  - should **start** in the next sprint?
  - performed in the past sprint should we **stop** and not introduce in the next sprint?
  - performed in the past sprint should we **continue** in the next sprint?



# Sprint Activities: Sprint Retrospective

- How are these activities implemented/enforced in the next sprint?

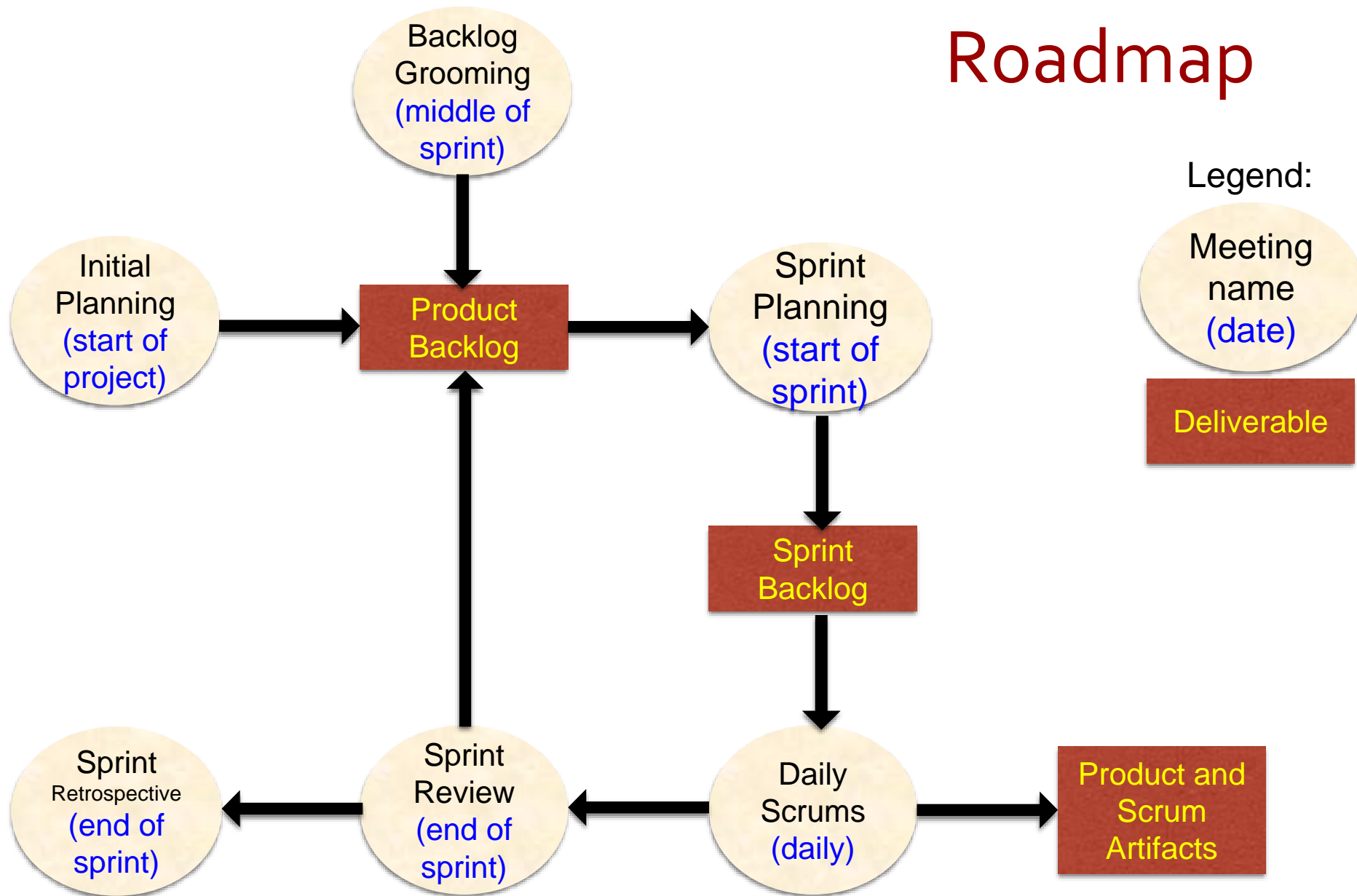
# Sprint Activities: Sprint Retrospective

- How are these activities implemented/enforced in the next sprint?
  - NB: Modification/Update of Definition of Done!!!
  - Adoption of new technologies
  - Redistribution of development roles if needed
    - e.g., Alice will switch from mostly DB work to front-end work

# Scrum: Sprint Meetings / Activities

- Initial Planning (start of the project)
- Sprint Planning Meeting (start of Sprint)
- Daily Scrum (daily)
- Backlog Grooming (~ middle of sprint)
- Sprint Review (end of Sprint)
- Sprint Retrospective (end of Sprint)

# Scrum Process Roadmap



# Scrum Omissions

# Scrum Omissions

- As a process model for new products (not just software), scrum omits many aspects of legacy software lifecycles

# Scrum Omissions

- Potential omissions include:

# Scrum Omissions

- Potential omissions include:
  - Programming!!!
  - Code Reviews
  - Static Analysis
  - Unit-Level Testing
  - Integration Testing
  - Beta Testing
- NB: Scrum doesn't prescribe what your project needs, team is responsible for these choices
  - hint: Definition of Done



# User Personas and Stories

# Getting to Know your User with User Personas

- Chances are... you may not know much about your user!

# Getting to Know your User with User Personas

- Chances are... you may not know much about your user!
- User Personas are fictional characters representing your target end users

# Getting to Know your User with User Personas

- Chances are... you may not know much about your user!
- User Personas are fictional characters representing your target end users

## User Persona Worksheet



Name: Wanda Bodine

Role: Sales Manager

Age: 39 (or so)

**Description:** Rhonda is in charge of 32 sales agents in the field. She is proud of being tech-savvy. Rhonda is hands-on with her data to a fault. she ends up working with Salesforce.com more than she interacts with her people. Rhonda is a level 68 shaman in World of Warcraft, and an accomplished amateur sushi chef.

Example from  
*Elements of Scrum*  
(pp.147)

# Getting to Know your User with User Personas

- Chances are... you may not know much about your user!
- User Personas are fictional characters representing your target end users
- One or two page description of their attitudes, needs, skills, work environment, challenges, etc.
- Often named... perhaps even a picture!

## User Persona Worksheet



Name: Wanda Bodine

Role: Sales Manager

Age: 39 (or so)

**Description:** Rhonda is in charge of 32 sales agents in the field. She is proud of being tech-savvy. Rhonda is hands-on with her data to a fault. She ends up working with Salesforce.com more than she interacts with her people. Rhonda is a level 68 shaman in World of Warcraft, and an accomplished amateur sushi chef.

Example from  
*Elements of Scrum*  
(pp.147)

# Developing for User Personas

- Have a **primary persona**, your product's target user
- Write User Stories for each of your personas
  - As a **Sales Manager**, I need to...

# Scenarios where Developing for Personas will be helpful

# Scenarios where Developing for Personas will be helpful

- Software design will promote common features used by these primary users
  - building an installer feature for an end-user vs. system admin.
- Thinking about personas could avoid designing unintuitive / hard to use software interfaces:



# Scenarios where Developing for Personas will be helpful

- Software design will promote common features used by these primary users
  - building an installer feature for an end-user vs. system admin.
- Thinking about personas could avoid designing unintuitive / hard to use software interfaces:
  - Blackboard

# Scenarios where Developing for Personas will be helpful

- Software design will promote common features used by these primary users
  - building an installer feature for an end-user vs. system admin.
- Thinking about personas could avoid designing unintuitive / hard to use software interfaces:
  - Blackboard
  - MS Office 2003
    - survey of Office 2003
    - >90% of users asked for features already available in Office (which had over 1,500 commands)
      - useful only for developers and super power-users

# Acceptance Criteria

# Acceptance Criteria Purpose

- Confirm a User Story is complete and working as the customer expects
- Augment User Stories with the necessary details to implement them as the customer intends

# Acceptance Criteria Purpose

- Enhance Team's understanding of the stories
- Elicit the detailed knowledge required for **estimating stories** and in Sprint Planning
- Define the boundaries of a User Story

# Acceptance Criteria: Where do They Come From?

- You can get some from your **customer**
- But **customers notoriously omit assumptions**
  - team needs to **read between the lines** and **anticipate events/situations**
  - respond with **“What if ...?” scenarios**
- **Conversation**: Team will likely need to write most Acceptance Criteria and discuss them with the customer

# Good Acceptance Criteria

- Any detail closing a divergence between:
  - the Team's and the customers' product visions
  - the various Team members' product visions
- Any detail likely to be implemented incorrectly (“We need a test case for this to ensure we get it right...”)

# Acceptance Criteria Template

- CS471 encourages use of the ... template



# Acceptance Criteria Template

- CS471 encourages use of the **Given-When-Then** template
- **Given** some-context **when** some-event-occurs **then** this-should-result
- Example:
  - **Given** valid username and password entries on the login screen, **when** the user clicks Login, **then** the server sends a session cookie to the browser

# Acceptance Criteria Template

- CS471 encourages use of the **Given-When-Then** template
- **Given** some-context **when** some-event-occurs **then** this-should-result
- Example:
  - **Given** valid username and password entries on the login screen, **when** the user clicks Login, **then** the server sends a session cookie to the browser
    - Q: What language is this acceptance criteria written in?
    - A:

# Acceptance Criteria Template

- CS471 encourages use of the **Given-When-Then** template
- **Given** some-context **when** some-event-occurs **then** this-should-result
- Example:
  - **Given** valid username and password entries on the login screen, **when** the user clicks Login, **then** the server sends a session cookie to the browser
    - Q: What language is this acceptance criteria written in?
    - A: Engineering Language (“session cookie”)

# Acceptance Criteria Language

- Industry guideline is to write Acceptance Criteria in the Customer's business language (just like User Stories)
- IMHO...
  - You definitely need Business Language when discussing Acceptance Criteria with the Customer
  - But... you may discover details in those discussions that can only be expressed in Engineering Language

# Acceptance Criteria Language

- The previous example (in **engineering language**):
  - *Given* valid username and password entries on the login screen,
  - *when* the user clicks Login,
  - *then* the server sends a session cookie to the browser

# Acceptance Criteria Language

- The previous example (in **engineering language**):
  - *Given* valid username and password entries on the login screen,
  - *when* the user clicks Login,
  - *then* the server sends a session cookie to the browser
- could have been written as (in **business language**):

# Acceptance Criteria Language

- The previous example (in **engineering language**):
  - *Given* valid username and password entries on the login screen,
  - *when* the user clicks Login,
  - *then* the server sends a session cookie to the browser
- could have been written as (in **business language**):
  - *Given* a valid username-password pair,
  - *when* the user clicks OK,
  - *then* the server will log them in

# Acceptance Criteria Language

- The previous example (in **engineering language**):
  - **Given** valid username and password entries on the login screen,
  - **when** the user clicks Login,
  - **then** the server sends a session cookie to the browser
- could have been written as (in **business language**):
  - **Given** a valid username-password pair,
  - **when** the user clicks OK,
  - **then** the server will log them in
- If you need it, **Engineering Language captures more detail**



# Engineering Language Captures more Details

- Consider the security issues associated with that session cookie...
- Written in Business Language
  - *Given* a valid username-password pair, *when* the server logs a user in, *then* no attacker will be successful at retrieving information

# Engineering Language Captures more Details

- Consider the security issues associated with that session cookie...
- Written in Business Language
  - **Given** a valid username-password pair, **when** the server logs a user in, **then** no attacker will be successful at retrieving information
- Written in (almost) Business Language
  - **Given** a valid username-password pair, **when** the server logs a user in, **then** no attacker will be able to hijack the session

# Engineering Language Captures more Details

- Written in Engineering Language
  - **Given** a successful login, **when** the server sends the session cookie, **then** the server uses HTTPS and tags the cookie HttpOnly
- Critical security details are in the Engineering Language and are not readily expressed in Business Language, nor does scrum capture them elsewhere

# Engineering Language Captures more Details

- Recommendation: Record critical engineering details in Acceptance Criteria unless they are obvious to all Developers
- When in doubt (“Should I specify or not?”) always specify