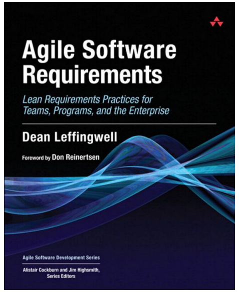
CE Department

Software Requirements Engineering

40688

These slides are designed to accompany Agile Software Requirements (2011) by Dean Leffingwell and support the university course Software Requirements Engineering, instructed by Mehran Rivadeh. Created and designed by Mahnaz Rasekhi.



Agile Software Requirements

Lean Requirements Practices for Teams, Programs, and the Enterprise

Dean Leffingwell

Foreword by Don Reinertsen

Agile Software Development Series

Alistair Cockburn and Jim Highsmith, Series Editors

Agile Software Requirements (2011)

Dean Leffingwell

User Stories_Part 3

Chapter 6

Mehran Rivadeh
mrivadeh@sharif.edu

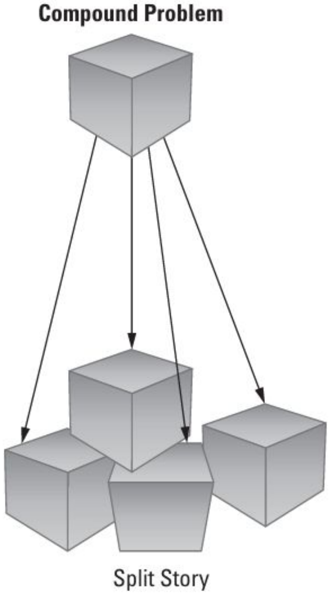
Software Requirements Engineering
October 2025 - Fall 1404 - SUT

Splitting User Stories

Contents

1. Workflow Step
2. Business Rule Variations
3. Major Effort
4. Simple/Complex
5. Variation In Data
6. Data Entry Method
7. Defer System Qualities
8. Operations
9. Use Case Scenarios
10. Break Out A Spike

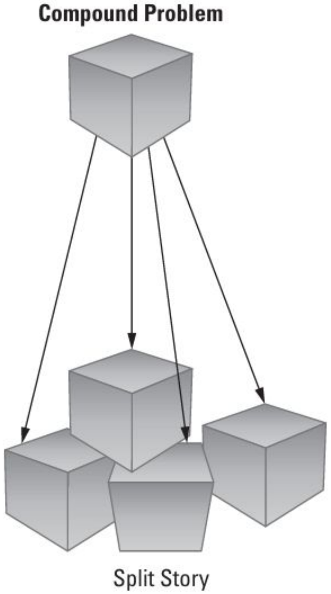
Splitting User Stories



User stories are often driven by epics and features—a large, vague concept of something we want to do for a user. We often find these big-value stories during our discovery process and capture them in the backlog.

However, these are compound stories, as pictured on the left, and are usually far too big to be implemented within an iteration. To prepare the work for iterations, a team must break them down into smaller stories.

Splitting User Stories



There is no set routine for splitting user stories into iteration-sized bites, other than the general guidance to make each story provide a vertical slice, some piece of user value, through the system.

However, we recommend applying an appropriate selection of ten common patterns to split a user story.

Splitting User Stories

Contents

1. **Workflow Step**
2. Business Rule Variations
3. Major Effort
4. Simple/Complex
5. Variation In Data
6. Data Entry Method
7. Defer System Qualities
8. Operations
9. Use Case Scenarios
10. Break Out A Spike

Workflow Step

- ➔ Identify specific steps that a user takes to accomplish a specific workflow.
- ➔ Implement the workflow in incremental stages.

Example:

- ➔ **As a utility, I want to update and publish pricing programs to my customer.**
  - ◆ This is a high-level goal.
  - ◆ It means the utility company wants to communicate pricing information (like electricity rates or time-of-use pricing) to customers using different channels.
  - ◆ It's too broad to implement in one go.

Workflow Step

Example (Cont.):

➔ As a utility, I want to update and publish pricing programs to my customer.

- . . . I can publish pricing programs to the customer’s in-home display.
- . . . I can send a message to the customer’s web portal.
- . . . I can publish the pricing table to a customer’s smart thermostat.

Splitting User Stories

Contents

1. Workflow Step
2. **Business Rule Variations**
3. Major Effort
4. Simple/Complex
5. Variation In Data
6. Data Entry Method
7. Defer System Qualities
8. Operations
9. Use Case Scenarios
10. Break Out A Spike

Business Rule Variation

- At first glance, some stories seem fairly simple.
- However, sometimes the business rules are more complex or extensive than the first glance revealed.
- In this case, it might be useful to break the story into several stories to handle the business rule complexity.

Example:

- **As a utility, I can sort customers by different demographics.**
  - At first glance, this seems simple—just sorting data.
  - But when you dig deeper, each type of sorting may involve different logic, data sources, and validation rules.

Business Rule Variation

Example (Cont.):

➔ As a utility, I can sort customers by different demographics.

- **Sort by ZIP code**
  - Requires accurate address data.
  - May need to handle ZIP code ranges or regional groupings.
  - Might involve filtering by service areas.
- **Sort by home demographics**
  - Involves data like home size, type (apartment, house), number of occupants.
  - May require integration with census or customer-provided data.
  - Privacy rules might apply.

Business Rule Variation

Example (Cont.):

➔ As a utility, I can sort customers by different demographics.

- **Sort by energy consumption**
    - Depends on historical usage data.
    - May involve thresholds (e.g., high vs low usage).
    - Could trigger alerts or recommendations.

Business Rule Variation

Example (Cont.):

- **As a utility, I can sort customers by different demographics.**
  - As a utility, I can sort customers by ZIP code, so that I can target regional programs.
  - As a utility, I can sort customers by home demographics, so that I can tailor energy-saving recommendations.
  - As a utility, I can sort customers by energy consumption, so that I can identify high-usage households for outreach.

Splitting User Stories

Contents

1. Workflow Step
2. Business Rule Variations
3. **Major Effort**
4. Simple/Complex
5. Variation In Data
6. Data Entry Method
7. Defer System Qualities
8. Operations
9. Use Case Scenarios
10. Break Out A Spike

Major Effort

- ➔ Sometimes, a user story requires building a foundation or infrastructure before additional features can be added.
- ➔ The first story takes the bulk of the effort, while the rest are easier to implement once that foundation is in place.

Example:

- ➔ **As a user, I want to be able to select/change my pricing program with my utility through my web portal.**
    - ◆ This is a broad capability.
    - ◆ It means the user wants to interact with their utility company online to choose how they're billed for energy.

Major Effort

Example (Cont.):

- ➔ As a user, I want to be able to select/change my pricing program with my utility through my web portal.
- ➔ But this requires building a system that:
  - ◆ Connects the web portal to backend pricing systems
  - ◆ Validates user eligibility
  - ◆ Handles enrollment and updates
  - ◆ This is the major effort—building the core infrastructure

Major Effort

Example (Cont.):

- ➔ As a user, I want to be able to select/change my pricing program with my utility through my web portal.
  - ◆ As a user, I want to use time-of-use pricing, so that I can save money by shifting my energy use to off-peak hours.
  - ◆ As a user, I want to prepay for my energy, so that I can manage my budget more easily.
  - ◆ As a user, I want to enroll in critical-peak pricing, so that I can reduce usage during high-demand periods and earn incentives.

Splitting User Stories

Contents

1. Workflow Step
2. Business Rule Variations
3. Major Effort
4. **Simple/Complex**
5. Variation In Data
6. Data Entry Method
7. Defer System Qualities
8. Operations
9. Use Case Scenarios
10. Break Out A Spike

Simple / Complex

- When the team is discussing a story and the story seems to be getting larger and larger
  - “What about x?”
  - “Have you considered y?”
- Stop and ask, “What’s the simplest version that can possibly work?”
- Capture that simple version as its own story.
- Break out all the variations and complexities into their own stories.

Example

- As a user, I basically want a fixed price, but I also want to be notified of critical-peak pricing events.

Simple / Complex

Example (Cont.):

- ➔ As a user, I basically want a fixed price, but I also want to be notified of critical-peak pricing events.
  - ➔ This starts simple—fixed pricing—but quickly adds layers:
    - ◆ Notifications for critical-peak events
    - ◆ Responding to time and duration of those events
    - ◆ Handling emergency events
  - ➔ Each of these adds complexity in terms of logic, data, and user interaction.

Simple / Complex

Example (Cont.):

- ➔ **As a user, I basically want a fixed price, but I also want to be notified of critical-peak pricing events.**
  - ◆ As a user, I want to be notified of critical-peak pricing events, so that I can adjust my energy usage.
  - ◆ As a user, I want to respond to the time and duration of critical-peak pricing events, so that I can optimize my energy savings.
  - ◆ As a user, I want to respond to emergency pricing events, so that I can help reduce grid stress.

Splitting User Stories

Contents

1. Workflow Step
2. Business Rule Variations
3. Major Effort
4. Simple/Complex
5. **Variation In Data**
6. Data Entry Method
7. Defer System Qualities
8. Operations
9. Use Case Scenarios
10. Break Out A Spike

Variation In Data

- The same feature must handle different types of data, formats, or sources.
- Each variation introduces new logic, complexity, or testing needs.
- You start with a simple version, then add variations as separate stories.

Example:

- **As a utility, I can send messages to customers.**
  - This is the simplest version. It assumes:
    - One language (likely English)
    - One message format
    - One delivery method

Variation In Data

Example (Cont.):

➔ **As a utility, I can send messages to customers.**

- As a utility, I can send messages to customers in Spanish, so that Spanish-speaking users can understand the communication.
- As a utility, I can send messages to customers in Arabic, so that Arabic-speaking users are included.
- As a utility, I can send messages to customers in other preferred languages, so that all users receive messages in their native language.

Variation In Data

Example (Cont.):

➔ **As a utility, I can send messages to customers.**

- As a utility, I can send messages to customers in Spanish, so that Spanish-speaking users can understand the communication.
- As a utility, I can send messages to customers in Arabic, so that Arabic-speaking users are included.
- As a utility, I can send messages to customers in other preferred languages, so that all users receive messages in their native language.

Variation In Data

Example (Cont.):

- Now each variation may require:
    - Translation workflows
    - Language detection or user preferences
    - Formatting adjustments (e.g., right-to-left for Arabic)

Splitting User Stories

Contents

1. Workflow Step
2. Business Rule Variations
3. Major Effort
4. Simple/Complex
5. Variation In Data
6. **Data Entry Method**
7. Defer System Qualities
8. Operations
9. Use Case Scenarios
10. Break Out A Spike

Data Entry Method

- ➔ Sometimes complexity is in the user interface rather than the functionality itself.
- ➔ In that case, split the story to build it with the simplest possible UI, and then build the richer UI later.

Example:

- ➔ As a user, I can view my energy consumption in various graphs.
  - ◆ This is a broad story.
  - ◆ The backend logic (retrieving and calculating consumption data) might be the same, but the way it's presented to the user can vary in complexity.

Data Entry Method

Example (Cont.):

- ➔ As a user, I can view my energy consumption in various graphs.
  - **Simple UI First**
    - As a user, I can view my energy consumption in a basic line chart, so that I can see how my usage changes over time.
      - This is the simplest version.
      - It delivers immediate value with minimal UI effort.

Data Entry Method

Example (Cont.):

- As a user, I can view my energy consumption in various graphs.
  - **Add Bar Charts**
    - As a user, I can view my energy consumption in bar charts that compare weekly usage, so that I can spot trends and spikes.
      - Adds a more visual and comparative element, but still relatively simple.

Data Entry Method

Example (Cont.):

- As a user, I can view my energy consumption in various graphs.
  - **Add Demographic Comparison**
    - As a user, I can view a comparison chart that shows my usage alongside similar households, so that I can understand how I compare to others.
      - This adds complexity: it requires demographic data, comparison logic, and a more advanced chart.

Splitting User Stories

Contents

1. Workflow Step
2. Business Rule Variations
3. Major Effort
4. Simple/Complex
5. Variation In Data
6. Data Entry Method
7. **Defer System Qualities**
8. Operations
9. Use Case Scenarios
10. Break Out A Spike

Defer System Qualities

- Sometimes, the initial implementation isn’t all that hard, and the major part of the effort is in making it fast or reliable or more precise or more scalable.
- However, the team can learn a lot from the base implementation, and it should have some value to a user, who wouldn’t otherwise be able to do it all.
- In this case, break the story into successive “ilities.”

Example:

- As a user, I want to see real-time consumption from my meter.
  - . . . interpolate data from the last known reading
  - . . . display real-time data from the meter.

Splitting User Stories

Contents

1. Workflow Step
2. Business Rule Variations
3. Major Effort
4. Simple/Complex
5. Variation In Data
6. Data Entry Method
7. Defer System Qualities
8. **Operations**
9. Use Case Scenarios
10. Break Out A Spike

Operation

- Words like manage or control are a giveaway that the story covers multiple operations, which can offer a natural way to split the story.

Example:

- **As a user, I can manage my account.**
  - . . . I can sign up for an account.
  - . . . I can edit my account settings.
  - . . . I can cancel my account.
  - . . . I can add more devices to my account.

Splitting User Stories

Contents

1. Workflow Step
2. Business Rule Variations
3. Major Effort
4. Simple/Complex
5. Variation In Data
6. Data Entry Method
7. Defer System Qualities
8. Operations
9. **Use Case Scenarios**
10. Break Out A Spike

Use Case Scenarios

- If use cases have been developed to represent complex user-to-system or system-to-system interaction, then the story can often be split according to the individual scenarios of the use case.

Example:

- **I want to enroll in the energy savings program through a retail distributor.**
  - Use case/story #1 (happy path): Notify utility that consumer has equipment.
  - Use case/story #2: Utility provisions equipment and data and notifies consumer.
  - Use case/story #3 (alternate scenario): Handle data validation errors.

Splitting User Stories

Contents

1. Workflow Step
2. Business Rule Variations
3. Major Effort
4. Simple/Complex
5. Variation In Data
6. Data Entry Method
7. Defer System Qualities
8. Operations
9. Use Case Scenarios
10. **Break Out A Spike**

Break Out A Spike

- ➔ In some cases, a story may be too large or overly complex, or perhaps the implementation is poorly understood.
- ➔ In that case, build a technical or functional spike to figure it out; then split the stories based on that result. (See the “Spikes” section_Chapter 6 - Part 1)

Contributions

- ➔ Author of Reference Book: **Dean Leffingwell**
- ➔ Course Instructor: **Mehran Rivadeh**
- ➔ Slide Creator: **Mahnaz Rasekhi**
  - ◆ These slides are primarily based on Agile Software Requirements by Dean Leffingwell, with occasional adaptations to enhance clarity and engagement.