Explore Scope with the DA Toolkit

Choose Your WoW!
A Disciplined Agile Delivery Handbook for Optimizing Your Way of Working

Foreword by Jonathan Smart

Mark Lines, CDAC, CDAI DA Fellow
Agenda

• Why you need to use multiple strategies for exploring scope
• Choosing the right level of detail
• Managing your work items
• Overview of scope modeling strategies
• Q&A
# The Process Goals of Disciplined Agile Delivery

<table>
<thead>
<tr>
<th>Inception</th>
<th>Construction</th>
<th>Transition</th>
<th>Ongoing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Form Team</strong></td>
<td><strong>Prove Architecture Early</strong></td>
<td><strong>Ensure Production Readiness</strong></td>
<td><strong>Ongoing</strong></td>
</tr>
<tr>
<td><strong>Align with Enterprise Direction</strong></td>
<td><strong>Address Changing Stakeholder Needs</strong></td>
<td><strong>Deploy the Solution</strong></td>
<td>Improve and work in an enterprise aware manner.</td>
</tr>
<tr>
<td><strong>Explore Scope</strong></td>
<td><strong>Produce a Potentially Consumable Solution</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Identify Architecture Strategy</strong></td>
<td><strong>Improve Quality</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Develop Release Plan</strong></td>
<td><strong>Accelerate Value Delivery</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Develop Test Strategy</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Develop Common Vision</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Secure Funding</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Inception**
Get the team going in the right direction.

**Construction**
Incrementally build a consumable solution.

**Transition**
Release the solution into production.

**Ongoing**
Improve and work in an enterprise aware manner.
Why do you need different strategies for Exploring Scope?

• Build a common understanding around
  – At what level of detail do we need to capture requirements?
  – How to we organize, and prioritize them?
  – What types of requirements models make sense for us
• DA Principles: #contextcounts #choiceisgood #pragmatism
Goal: Explore Scope

Explore Usage

Explore the Domain
- Domain/conceptual model
- Glossary
- Logical Data Model (LDM)
- UML class diagram

Explore the Process
- Business process diagram
  - Data flow diagram (DFD)
  - Flow chart
  - UML activity diagram
  - UML state chart
  - Value stream map

Explore User Interface (UI) Needs
- User interface (UI) flow diagram
  - UI prototype (high fidelity)
  - UI prototype (low fidelity)
  - UI specification

Explore General Requirements
- Business rule
- Context diagram
- Feature statements
- Impact map
- Mind map
- Shall statements

Explore Quality Requirements

Apply Modeling Strategy(ies)

Choose a Work Item Management Strategy

Agile modeling (informal) sessions
- Open space
- Joint Application Requirements (JAR) sessions
- Interviews

Work item pool
- Taskboard
- Work item list

Acceptance criteria
- Explicit list
- Technical stories

Outcome driven
- Requirements envisioning (light specification)
- Detailed specification
- No models

Epic
- Outcome
- Persona
- UML use case diagram
- Usage scenario
- Use case
- User story
- User story map
Each Option/Strategy is Defined, with Trade-offs in Choose Your WoW! which is the official DAD BOK
Level of Detail of the Scope Document

Outcome driven
Requirements envisioning (light specification)
Detailed specification
No models
Level of Detail of the Scope Document

→ **Requirements Envision (light specification)**

- A set of light-weight models, typically captured as sketches and minimal text descriptions (such as those described by Agile Modeling). Examples include user stories, personas, story maps, low-fidelity UI prototypes.

---

### Mary
- **Age:** 37
- **Single,** 2 daughters 11 and 9 years old
- **Role:** Customer for 3 years, has and existing CI policy

**Key Attributes:**
Mary is recently divorced and has retained the family restaurant. She needs to change her insurance coverage and look for lower premium alternatives now that she has only one income.

**Goals:**
- Early renewal of policy with flexible premium alternatives.

### John
- **Age:** 57
- **Married,** 1 son 32 years old
- **Role:** Underwriter, has worked for LM for 15 years

**Key Attributes:**
John is conservative by nature. He has used the old system for over 8 years. Even though the existing system is inflexible, he knows how to get what he needs from it. He is skeptical that a new system would be better than what he has.

**Goals:**
- Improved system that is easy to use as his current system with a painless migration.

---

**Scenario:** Sufficient availability in Seminar

*Given* a Student is qualified to enroll

*And* the Seminar has availability

*When* we determine whether to allow enrollment

*Then* the Student should be enrolled

© Disciplined Agile, Inc.
Level of Detail of the Scope Document ➔ No Models

• The stakeholders describe their needs to the team and the team produces something based on that conversation.
Explore Usage

- Epic
- Outcome
- Persona
- UML use case diagram
- Usage scenario
- Use case
- **User story**
- **User story map**
Explore Usage → Epic

- Large stories that take a lot of effort, often multiple iterations, to complete. Epics are typically organized into a collection of smaller user stories. Sometimes Epics are referred to as Features or User Activities.
## Explore Usage → Personas

Detailed descriptions of fictional people who fill roles as stakeholders of the solution being developed.

<table>
<thead>
<tr>
<th>Mary</th>
<th>John</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Mary" /></td>
<td><img src="image" alt="John" /></td>
</tr>
<tr>
<td><strong>Age:</strong> 37</td>
<td><strong>Age:</strong> 57</td>
</tr>
<tr>
<td><strong>Single, 2 daughters 11 and 9 years old</strong></td>
<td><strong>Married, 1 son 32</strong></td>
</tr>
<tr>
<td><strong>Role:</strong> Customer for 3 years, has existing CI policy</td>
<td><strong>Role:</strong> Underwriter, has worked for LM for 15 years</td>
</tr>
<tr>
<td><strong>Key Attributes:</strong></td>
<td><strong>Key Attributes:</strong></td>
</tr>
<tr>
<td>Mary is recently divorced and has retained the family restaurant. She needs to change her insurance coverage and look for lower premium alternatives now that she has only one income.</td>
<td>John is conservative by nature. He has used the old system for over 8 years. Even though the existing system is inflexible, he knows how to get what he needs from it. He is skeptical that a new system would be better than what he has.</td>
</tr>
<tr>
<td><strong>Goals:</strong> Early renewal of policy with flexible premium alternatives.</td>
<td><strong>Goals:</strong> Improved system that is easy to use as his current system with a painless migration.</td>
</tr>
</tbody>
</table>
Prioritizing Personas
– Who are you building your system for?

• Highest priority in the middle
• Two next priorities in the middle circle
• The rest in the outer circle

This helps us to manage usability tradeoffs between types of users
• Unified Modeling Language (UML) use case diagram. Diagrammatic notation for a textural use case.
Explore Usage ➔
Use Case

- Textural specification describing all different usage scenarios for the goals of the system

Use Case: UC1 Process Sale

1. *Brief Description*

   The Process Sale use case allows the Cashier to process a sale of groceries on behalf of the Customer. This use case includes scanning grocery items, applying coupons, and accepting payment for the transaction.

2. *Basic Flow*

   1. Cashier scans grocery item
   2. System displays item price on POS Terminal display
   3. System adds amount to total transaction cost
   4. For each additional item, return to Step 1 of the Basic Flow
   5. Cashier selects to conclude sale
   6. Cashier enters cash total provided by Customer
   7. System dispenses coin change
   8. Cashier provides bill change to the Customer
   9. Cashier closes drawer
   10. System prints receipt

© Disciplined Agile, Inc.
• One or two sentences to describe something of value to a user

Three common formats:
- Free form
- Short form: As a [Role] I want to [Something of value]
- Long form: As a [Role] I want to [Something of value] so that I can [Reason]

Students request official transcripts

As a Student I want to request an official transcript

As a Student I want to request an official transcript so that I can provide it to a potential employer
• User stories are placed on a flat surface (a wall in the case of sticky notes, a table in the case of index cards, or a screen in the case of digitally captured stories). They are then organized to indicate the epic they are part of and the production release they are assigned to.
Explore the Domain

- Domain/conceptual model
  - Glossary
  - Logical Data Model (LDM)
  - UML class diagram
Explore the Domain

→ Domain/Conceptual Model

- A high-level data model showing the entities and the relationship between them. Attributes of the entities are optionally indicated.
Explore the Process

- Business process diagram
  - Data flow diagram (DFD)
  - Flow chart
  - UML activity diagram
  - UML state chart
  - Value stream map
Explore the Process → Business Process Diagram

- Used to depict the activities and the logical flow between them within a process. Could be done in freeform format or with a notation such as Business Process Modeling Notation (BPMN)
Explore User Interface (UI) Needs

- User interface (UI) flow diagram
  - UI prototype (high fidelity)
  - UI prototype (low fidelity)
  - UI specification
Explore User Interface (UI) Needs → User Interface (UI) Flow Diagram

- Explores how the various screens and reports all fit together. Often created as a sketch on a whiteboard or with sticky notes on a drawing surface. Sometimes called a wireframe diagram.
Explore User Interface (UI) Needs → **UI Prototype** (low fidelity)

- Identify requirements for screens and reports using inclusive tools such as paper and whiteboards. Also called a screen sketch.
Explore User Interface (UI) Needs → UI Specification

Define exactly how a screen or report is to be built by the development team.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Label on the screen</td>
<td>if the element has no label, number it and refer to it by number</td>
</tr>
<tr>
<td>Description</td>
<td>Describe the element</td>
<td>its type (input, drop-down, calendar), what it does, etc.</td>
</tr>
<tr>
<td>Default value</td>
<td>what the field defaults to if no value is provided</td>
<td>May not be applicable for every type of screen</td>
</tr>
<tr>
<td>Values</td>
<td>List boundary conditions or error conditions</td>
<td>i.e., Dates must be in the past, integer numbers from 1 to 100</td>
</tr>
</tbody>
</table>
Explore General Requirements

- Business rule
- Context diagram
- Feature statements
- Impact map
- Mind map
- Shall statements
Explore Quality Requirements

Acceptance criteria

Explicit list

Technical stories

Explore Quality Requirements
Explore Quality Requirements → Acceptance Criteria

- Quality-focused approach that captures detailed aspects of a high-level requirement from the point of view of a stakeholder.

**Scenario:** Sufficient availability in Seminar

Given a Student is qualified to enroll
And the Seminar has availability
When we determine whether to allow enrollment
Then the Student should be enrolled

**Feature:** Enroll a Student in a Seminar

**Scenario:** Spot available
Given the Student is qualified to enroll in the Seminar
And the Seminar has availability
When the Student requests to enroll
And the Seminar has 5 spots remaining
Then the Student is enrolled in the Seminar
And the spots remaining should be 4

**Scenario:** No spots available
Given the Student is qualified to enroll in the Seminar
And the Seminar is full
When the Student requests to enroll
And the Seminar has 0 spots remaining
Then the Student is not enrolled in the Seminar
Explore Quality Requirements → Explicit List

- Enables us to capture quality requirements in a “reusable manner” that cross-cuts functional requirements.
Apply Modeling Strategy(ies)

Agile modeling (informal) sessions
- Open space
- Joint Application Requirements (JAR) sessions
- Interviews
Apply Modeling Strategy(ies) → **Agile modeling (informal) sessions**

- A light-weight, collaborative approach to modeling where stakeholders are often actively involved using simple, inclusive modeling tools such as whiteboards and paper
Choose a Work Item Management Strategy

Choose a Work Item Management Strategy

Work item pool
- Taskboard
- Work item list
- Requirements (product) backlog
- Unmanaged
Choose a Work Item Management Strategy → Work Item Pool

- A lean approach that enables team to implement several prioritization strategies simultaneously. Examples of prioritization strategies include business value, items to be expedited, fixed date, and intangible items such as paying down technical debt or attending training.

Work Item Pool

- Business Value
- Fixed Date
- Defects
- Learning
- Tech Debt
- Options

Work items are pulled when capacity is available to address them.
Choose a Work Item Management Strategy → Taskboard

• A lean strategy where the lifecycle, including prioritization, of work items is managed visually by the team.

<table>
<thead>
<tr>
<th></th>
<th>To Do</th>
<th>Doing</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Value</td>
<td><img src="image1" alt="To Do" /></td>
<td><img src="image2" alt="Doing" /></td>
<td><img src="image3" alt="Done" /></td>
</tr>
<tr>
<td>Fixed Date</td>
<td><img src="image4" alt="To Do" /></td>
<td><img src="image5" alt="Doing" /></td>
<td><img src="image6" alt="Done" /></td>
</tr>
<tr>
<td>Defects</td>
<td><img src="image7" alt="To Do" /></td>
<td><img src="image8" alt="Doing" /></td>
<td><img src="image9" alt="Done" /></td>
</tr>
<tr>
<td>Learning</td>
<td><img src="image10" alt="To Do" /></td>
<td><img src="image11" alt="Doing" /></td>
<td><img src="image12" alt="Done" /></td>
</tr>
<tr>
<td>Tech Debt</td>
<td><img src="image13" alt="To Do" /></td>
<td><img src="image14" alt="Doing" /></td>
<td><img src="image15" alt="Done" /></td>
</tr>
</tbody>
</table>
Choose a Work Item Management Strategy → Work Item List

- Similar to a Scrum product backlog but includes all types of work, not just requirements. In addition to value, work is also prioritized to implement risk-related items early.

Work Item List

Highest-Priority Work Items
Iteration Backlog

Iteration planning session to select work items and identify work tasks for current iteration.
Parting Thoughts

• If you are using User Stories and Acceptance criteria, you will likely need to supplement them with other models (textural & diagrammatic).

• DA Principle: Choice is good - Agile teams own their own process, they get to choose their Way of Working (WoW)

• DA Principle: Context counts – The modeling strategies you choose should suite your situation
Better decisions lead to better outcomes
Join Today

Sign up at DisciplinedAgileConsortium.org and become a member

Have access to new and upcoming information about Disciplined Agile strategies
The Disciplined Agile Framework