Pragmatic Product Leader Course

Become a Product Manager with Superpowers

LESSON 6B

Modeling with SDLC,OO & B.O.O.M

6.2. OO and SDLC.

SDLC and Modeling

Quick recap of SDLC:

1. **Initiation:** Make the business case for the project
2. **Discovery:** Conduct investigation leading to an understanding of the solution's desired behaviour
3. **Development:** Complete the analysis and design, code, integrate and test the software
4. **Final verification & validation:** Perform final testing before the product is transitioned into production
5. **Deployment:** Manage and coordinate deployment into production and close the product for now.

SDLC and Modeling: Initiation

Purpose: Get a rough cut at the business case
Trouble: Hardly any information, so difficult to estimate and hence difficult to justify requirement analysis.
Answer: Do *just enough* research to create a ballpark estimate

High-level needs:

- **Business use cases:** A tool for identifying and describing end-to-end business processes affected by the project.
- **Activity diagrams:** Used to help you and stakeholders form a consensus regarding the workflow of each business use case.
- **Actors:** These describe the users and external systems that will interact with the proposed IT system.
- **System use cases:** Used to help stakeholders break out the end-to-end business processes into meaningful interactions with the IT system.

Following is a list of the steps you'll carry out during this phase.

1. Model business use cases
    a. Identify business use cases (business use-case diagram)
    b. Scope business use cases (activity diagram)
2. Model system use cases
    a. Identify actors (role map)
    b. Identify system use-case packages (system use-case diagram)
    c. Identify system use cases (system use-case diagram)
3. Begin structural model (class diagrams for key business classes)
4. Set baseline (PRD/Initiation)

SDLC and Modeling: Discovery

Purpose: understand the solution's desired behavior and baseline the architecture

Remember: All the steps are not necessary

Main techniques to use in this phase:

- System use-case descriptions (specifications), storyboarding the interaction between users and the proposed IT system as each system use case is played out
- State-machine diagrams describing the lifecycle of key business objects
- Class diagrams describing key business concepts and business rules that apply to business objects such as accounts, investments, complaints, claims, and so on

Following is a list of the steps you'll carry out during this phase.

- Behavioral analysis
  - Describe system use cases (use-case description template)
  - Describe state behavior (state-machine diagram)
    - Identify states of critical objects
    - Identify state transitions
    - Identify state activities
    - Identify superstates
    - Identify concurrent states

contd'

SDLC and Modeling: Discovery

Purpose: understand the solution's desired behavior and baseline the architecture

- Structural analysis (object/data model) (class diagram)
  - Identify entity classes
  - Model generalizations
  - Model transient roles
  - Model whole-part relationships
  - Analyze associations
  - Analyze multiplicity
  - Link system use cases to the structural model
  - Add attributes
  - Add lookup tables
  - Distribute operations
  - Revise class structure
- Specify testing (test plan/decision tables)
  - Specify white-box testing quality level
  - Specify black-box test cases
  - Specify system tests
- Specify implementation plan (implementation plan)
- Set baseline for development (PRD/Discovery)

SDLC and Modeling: Development

Purpose: Support technical development

Business-analysis activity during this phase depends on the lifecycle approach being used.

On waterfall projects, where all the analysis is done up front, there is no requirements gathering or analysis during this phase; however, the PM is involved in supporting quality assurance and validating that the technical design meets the requirements (for example, by reviewing test plans and design specifications).

On iterative projects, where requirements analysis and solution development take place over a number of iterations, the steps described for the Discovery phase are carried out during each iteration of the Construction phase.

SDLC and Modeling: V&V

Purpose: Support testing

The Product Manager supports final testing before the completed solution is deployed, reviewing test plans and results and ensuring that all requirements are tested.

SDLC and Modeling: Closeout

Purpose: Support deployment

The business analyst supports the deployment process, reviewing transition plans and participating in a post-implementation review (PIR) to evaluate the success of the change.

What do you define first - Attributes or Operations?

- Encapsulation suggests that operations > attributes
- Context of business analysis, easier to identify the attributes of a class. The attributes show up as fields on screens and reports.
- Difficult to ascribe operations to classes
- No rule, start early and fill in the blanks. Consider B.O.O.M. as starting point. You will get to the end result - comprehensive requirements - relatively effortlessly, but over time, customize the process as you see fit.

Developing the structural model alongside behavioral model

- Behavioral and structural analysis should be performed in parallel
- For first timers, difficult to jump between the two

An approach:

- During the Initiation phase, identify system use cases in the behavioral model. Nouns discovered during this process are added to the structural model if they relate to new business concepts or objects. For example, the system use case "Adjudicate Loan Application" introduces the term Loan Application, which you define in the structural model. You continue working on the structural model during the Initiation phase, describing key business classes and their relationships to each other.
- Following the Initiation phase, as you describe each system use case, you verify it against the existing structural model. Does the system use case comply with rules expressed in the structural model? Has the system use case introduced new classes? You resolve any differences between the system use case and the structural model and update the structural model if necessary.
- By the time you have described the last system use case, the structural model should be complete and fully verified

Tailoring B.O.O.M. for your project

- B.O.O.M. steps are meant to be used as a checklist, not every step is necessary
- Guiding principle "If it isn't going to make a difference to the outcome, don't do it"

Degree of documentation and analysis require depends on a number of factors:

- Degree of formality versus adaptiveness of the lifecycle
  - Definitive or empirical lifecycle
  - For empirical lifecycle:
    - Do as little documentation as you can get away with
    - Do it as late in the process as possible
    - Don't baseline the requirements unless they are in the process of being implemented
- Iterative or sequential approach in SDLC
- Degree of uncertainty tolerated by the sponsor and the budget
- Regulatory requirements
- Size of team and physical distance between PM, solution team and business stakeholders
- Developer capability
- In-house or vendor solution
- Maturity of organisation

What do you show stakeholders?

Not all documents are appropriate for business stakeholders

Some artifacts to consider are:

- Activity (workflow) diagrams
- State-machine diagrams
- Use-case diagrams
- Use-case descriptions, decision tables and decision trees
- Class diagrams

Consider re-reading this slide once you've learned more about these artifacts

To conclude

Thank you.