Overview

- Developing Systems
  - System requirements
  - Design architecture
- Generic Process Context
  - Input requirements and derived requirements
  - Acceptance criteria and qualification strategy
- Generic Process Introduction
- Development in the context of change
- Generic Process Information Model
- Analyze & Model
- Derive Requirements and Qualification Strategy
Introduction

- Introduce the concept of process for the development of systems
- Examine the way in which systems are developed
- Identification of a development pattern
- Development pattern is expressed as a generic process
- Instantiation of process for specific purpose
Developing Systems

- Establish *need* for system before it is developed
- If purpose is not known, unclear what sort of system will be developed
- Impossible to determine whether system will satisfy needs:
  *If you don’t know where you are going, you are unlikely to get there. (Forest Gump)*
- Stakeholder requirements: transform vague statement of need into a set of needs as basis for purchasing
- Stakeholder: people who directly interact with system and those who have interest in its existence (e.g., think of a fire department).
Developing Systems

▪ Development process visualization:
  - Circles or ovals represent processes
  - Rectangles represent data or information that is read or produced
  - Arrow indicate data read or write process
  - Grey indicates concurrent work on requirements

▪ E.g., *Develop Stakeholder Requirements* takes *statements of needs* and produces *stakeholder requirements* while also creating and reading the *use model*
Once stakeholder requirements exist Systems Engineer can start thinking about possible solutions

- First, determine what characteristics system must have (rather than jumping straight to design)
- First, determine characteristics of system irrespective of final detailed design
- Produce an abstract model of the system.
- Model provides basis for discussion and common understanding of the proposed solution.
Example: High-level CASA Model

gap - earth curvature prevents 72% of the troposphere below 1 km from being observed.
Example: High-level CASA Model

NEXRAD = 5 min between updates
CASA = 1 min between updates
Developing Systems: Design Architecture

- Design architecture is expressed as set of interacting components collectively exhibiting desired properties

- Defines
  - What system components must do
  - How they interact with each other

- Defines requirements for each system component in terms of:
  - Functionality
  - Interaction

- Stipulate required properties: physical size, performance, reliability, maintainability
Generic Process Context

Statement of Needs

Stakeholder Requirements

System Requirements

System Components Requirements

Engineer Requirements

Engineer Requirements

Engineer Requirements

Engineer Requirement

Engineer Requirements

Subsystem Components Requirements

Problem Domain

Solution Domain

ECE Department
Generic Process Context

- Significant degree of commonality at each level
- Explore common aspects to present generic process
- Each level of development demands relevant expertise
- People or organizations with proven track record in the development of similar systems are necessary
- Subsystem developers bring own domain experience
Input & Derived Requirements

- Statement of Needs
  - Engineer Requirements
  - Stakeholder Requirements
- Engineer Requirements
- Stakeholder Requirements
- Subsystem Requirements
- Subsystem Requirements
- Engineer Requirements
- Derived Requirements

ECE Department
Acceptance Criteria & Qualification Strategy

- Information concerning the qualification strategy for requirements
- Consider how requirements will be demonstrated when system has been implemented
- Qualification strategy:
  - Testing
  - Trials
  - Certification
  - Inspection
- E.g., in avionics assure that flight instruments will perform to a known standard before first flight
Acceptance Criteria & Qualification Strategy

- Input Requirements
- Qualification Strategy for Input Requirements
- Derived Requirements
- Qualification Strategy for Derived Requirements
- Engineer Requirements
Generic Process Introduction: Ideal Development

- Process commences with need to agree input information for project with customer at level above
- Analyze input information and consider how to develop required outputs
Generic Process Introduction: Ideal Development

- Input Requirements
- Agree Requirements
- Qualification Strategy for Input Requirements
- Analyze & Model
- Model
- Analysis Results
- Engineer Requirements
- Derived Requirements
- Agree Requirements
- Qualification Strategy for Derived Requirements
Development in the Context of Change

- At early stage change can and must be made with ease to allow progress
- Later more formal arrangement is required
- Changes are first requested and then decision is made based on impact on project
- Will involve Program Manager and change control board.

Flow for changes is usually upwards:
- Limitation in a model or anomaly in analysis
- Discovered whilst attempting to generate derived requirement
- Will result in modification of model and/or additional analysis
Development in the Context of Change

- Input Requirements
- Agree Requirements
- Qualification Strategy for Input Requirements
- Derived Requirements
- Agree Requirements
- Qualification Strategy for Derived Requirements

Change Request

Engineer Requirements

Analyze & Model

Model

Analysis Results

Change Request

Change Request

Change Request
Before considering subprocesses, useful to introduce generic information model
Mainly diagrams used for information model
Purpose is to:
• Indicate what types of information exists
• Whether relationships exist between items of information
State transition diagram to indicate how state of each type of information can be changed
• Visual indication when and how processes interact with each other
### Information Classes

<table>
<thead>
<tr>
<th>Input Requirement</th>
<th>Qualification Strategy for Input Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agreement State</td>
<td>Agreement State</td>
</tr>
<tr>
<td>Qualification State</td>
<td></td>
</tr>
<tr>
<td>Satisfaction State</td>
<td></td>
</tr>
</tbody>
</table>

- Derived Requirement
  - Agreement State
  - Qualification State
  - Satisfaction State
  
  **Qualifies** for

- Qualification Strategy for Derived Requirement
  - Agreement State

**Qualifies** by

**Details for** by

**Imposed by**

**Impacted by**

**May impact**

**Change Request**
Agreement State

- Represents state of single requirement at some point in its history
- \textit{Being assessed} is super state that contains other states
- \textit{Proposed state}: Requirement is sufficiently well stated to be sent to supplier
- \textit{Being assessed state}: Customer and supplier negotiate until agreed requirement
- \textit{Agreed state}: Requirement stays there until change request is created
Agreement State

Customer assessing requirement from supplier

Supplier assessing requirement from customer

Proposed

Requirements proposed to supplier

Change from supplier

Alternative proposal from supplier

Acceptable

Requirements acceptable

Agreed

Alternative proposal from customer

Change from customer
Qualification State

No Qualification Strategy decided

Verification criteria agreed

Qualification Strategy decided

Change proposed

No Qualification Strategy suspect

Change **impacts** Qualification Strategy

Change **does not impact** Qualification Strategy
Satisfaction State

- Not satisfied
  - Requirement satisfied
  - Satisfied
    - Change proposed
    - Satisfaction suspect
      - Change impacts lower level supplier

Change **does not impact** lower level supplier
Information Model Constraints

- Change request binds together:
  - Agreement,
  - Qualification, and
  - Satisfaction state

- Registering a change request immediately changes all three states

- Work involved to determine impact and address consequences

- Agreement state of derived requirement must be consistent with state of input requirement \( \Leftarrow \). Input requirement cannot achieve satisfied state until lower level supplier has agreed all of derived requirement that satisfy it
Generic Process Details: Agreement Process

- Concurrent process between supplier and customer
- Before start of derivation work, *input requirements* need to be assessed
  ➔ ascertain they form adequate basis for development process
- Assessment must answer following questions:
  - Is the requirement complete?
  - Is the requirement clear?
  - Is the requirement implementable?
  - Is the qualification plan clear and acceptable?
Answers that lead to requirement rejection are:
- Missing information – e.g., “TBA”, “TBC”, “TBD”
- Lack of clarity – ambiguity, contradiction, confusion, etc.
- Impossible to implement – no known solution
- Unacceptable qualification plan

Often, acceptance criteria and qualification plans are decided fairly late:
- E.g., well after requirements have been agreed
- Sometimes prior to testing

Bad practice!! Leads to delays caused by late changes in requirements to make them testable
Generic Process Details: Agreement Process

- Derive Requirements & Qualification Strategy
  - Change Request
  - Higher Level Responsibility
  - Derived Requirements
  - Input Requirements
  - Change Request/Proposal from Supplier
  - Change Request/Proposal from Customer
  - Agree Derived Requirements & Qualification Strategy
  - Qualification Strategy for Derived Requirements
  - Qualification Strategy for Input Requirements
  - Lower Level Responsibility

ECE Department

Analyze & Model
Analyze and Model

- Analysis: Understanding nature and scope of input requirements to assess risk involved in satisfying them
- Analysis work:
  - Feasibility studies
  - Exploration of potential implementation options
  - Building of prototypes
- Build performance models to investigate throughput and response figures
Analyze and Model

- Models: Understand nature of and derive structure for derived requirements

- Models:
  - Use cases or user scenarios to understand stakeholder requirements
  - Design architectures to identify elements of solution and indicates their interaction

- Model is used to establish design architecture of proposed solution
  - Obvious for well-established development domains
  - No established architecture for innovative developments
## Analyze & Model: Modeling Techniques

<table>
<thead>
<tr>
<th>Industry</th>
<th>Models</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Aircraft Industry</strong></td>
<td>Aerodynamic model, Three-dimensional spatial model, Weight distribution model, Flight simulator</td>
</tr>
<tr>
<td><strong>Rail Industry</strong></td>
<td>Timetable simulation, Safety, reliability and maintainability models</td>
</tr>
<tr>
<td><strong>Car Industry</strong></td>
<td>Styling model, Dashboard model, Aerodynamic model</td>
</tr>
</tbody>
</table>
Analyze and Model

- Objective of models:
  - Understand *input requirements* together with
  - Proposed qualification strategy and
  - Experiment with alternative solution options prior to
  - Deciding how to proceed with creation of *derived requirements*

- During Analyze & Model process:
  - Likely further questions will arise concerning meaning and formulation of *input requirements*
  - Gives rise to change requests, which causes *agree requirements* process to be re-entered
Analyze & Model

- Input Requirements
- Change Request
- Analyze & Model
- Model
- Analysis Results
- Change Request

Derive Requirements & Qualification Strategy

Agree IR & IQS

Qualification Strategy for Input Requirements
Derive Requirements and Qualification Strategy

- Derivation of component requirements based on design architecture
- Determine specific requirements that must be satisfied by each component
- Requirements consist of constraints imposed either by component architecture or input requirements
- Constraints include:
  - Interface constraints
  - Physical constraints such as mass, volume, power usage, heat dissipation, etc.
Derive Requirements and Qualification Strategy

Input Requirements → Analyze & Model → Engineer Requirements → Change Request → Derived Requirements

Model → Change Request → Analysis Results → Qualification Strategy for Input Requirements

Derivation & Qualification for Derived Requirements → Agree DR & DQS
Derive Requirements and Qualification Strategy

- Consists of a set of qualification actions:
  - Trial
  - Test
  - Inspection

- Qualification action should take following aspects into account:
  - The *kind* of action that would be appropriate for the requirement
  - The *stage* at which each action could take place – the earlier the better
  - Any special
The vessel shall be capable of traveling at 40 knots while in up to sea condition D.

The hull shall present a maximum drag coefficient of X.

The engine shall deliver a minimum thrust of Y.

System Qualification Strategy:
- Sea trial of vessel in sea condition A
- Sea trial of vessel in sea condition D

Subsystem Qualification Strategy:
- Drag coefficient tests using a pre-build scale model
- Use of factory test rig to measure power output
Find configuration that optimizes utility at time step \( k \):

\[
J = \max_{\text{configurations}, C} \sum_{\text{tasks}, t} U(t, k) Q(t, C)
\]

Utility – “how important” is task \( t \) to the users at time \( k \)?

\[
U(t, k) = \sum_{\text{groups}, g} w_g U_g(t, k)
\]

Quality – “how good” is scanning configuration \( C \) (distance, coverage, # radars) for task \( t \)?
Find \textit{configuration} that optimizes utility at time step \( k \):

\[
J = \max_{\text{configurations}, C} \sum_{\text{tasks}, t} U(t, k) Q(t, C)
\]

- separation of “how \textit{important},” \( U(t, k) \), from “how \textit{good},” \( Q(t, C) \)
- \( U(t, k, Q(t, C)) \) would have been possible but:
  - complex to solve
  - complex to specify and update \( U(t, k, Q(t, C)) \)
  - “stovepipe” design
user values for detected weather features

<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Prior Information available</th>
<th>NWS utility Wt=0.4</th>
<th>EM utility Wt=0.3</th>
<th>Researcher utility Wt=0.2</th>
<th>Vieux utility Wt=0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TVS detection</td>
<td>AOP</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Remote</td>
<td></td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Mesocyclone</td>
<td>AOP</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Remote</td>
<td></td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Storm cell</td>
<td>AOP</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>
CASA Example

• “naturally”: group-sensitive utility for each feature (tornado, wind shear, hail core) scanned

• ... and the survey says....

User feedback:

• NWS: want “mental movie” scanning “areas of interest” at regular intervals
• feature-based too “jumpy”
• need context: scan areas around features (storm cell)
• dynamic data requests
**CASA Example**

- *interval-based preferences:* “do X every Y time units”

<table>
<thead>
<tr>
<th>Rules</th>
<th>Rule trigger</th>
<th>Sector Selection</th>
<th>Elevations</th>
<th># radars</th>
<th>Contig.</th>
<th>Sampling interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>NWS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N1</td>
<td>time</td>
<td>360</td>
<td>lowest</td>
<td>1</td>
<td>Yes</td>
<td>1 / min</td>
</tr>
<tr>
<td>N2</td>
<td>storm</td>
<td>task size</td>
<td></td>
<td>1</td>
<td>Yes</td>
<td>1 / 2.5 min</td>
</tr>
<tr>
<td>EMs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>time</td>
<td>360</td>
<td>lowest</td>
<td>1</td>
<td>Yes</td>
<td>1 / min</td>
</tr>
<tr>
<td>E2</td>
<td>reflectivity over AOI</td>
<td>task size</td>
<td></td>
<td>1</td>
<td>Yes</td>
<td>1 / min</td>
</tr>
<tr>
<td>E3</td>
<td>velocity over AOI</td>
<td>task size</td>
<td>lowest</td>
<td>2+</td>
<td>Yes</td>
<td>1 / 2.5 min</td>
</tr>
</tbody>
</table>