Albert Kregl

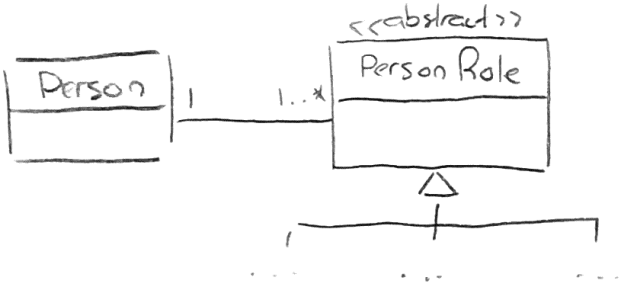
ECSE 326 – Cheat Sheet™

- **Vision Statement:** "For [target customer] who [need or opportunity], the [product name] is [product category] that [key benefit], unlike [main competitive alternative] our product [statement of main differentiation]."
- **Requirements Engineering Activities:** Inception, Elicitation, Analysis, Specification, V&V, Management
- **3 Categories of NFRs:** Performance, Design Constraints, Commercial Constraints
- **Requirements Levels:** Business Rules, User Requirements, System Requirements
- **Problem Statement:** "The problem of ___ affects ___, the impact of which is ___. A successful solution would be ___."
- **Inception:** Define product vision, product scope
- **Elicitation:** "Discover" requirements by communicating with customers and system users
- **Elicitation Techniques:** Interviews, Brainstorming, Existing System Analysis, Questionnaires, Prototyping, Observation (Artifact, stakeholder, model, creativity, data-based)
- **Sources of Requirements:** Stakeholders, system environment, current problems, previous systems, existing docs, requirements checklists
- **IEEE 830-1998:** Recommended practice for Software Requirements Specifications, describes the qualities and content of good SRSs.
- **IEEE 29148:2011:** More emphasis on characteristics of good requirements, RE activities and processes, operations. Harmonizes IEEE 830, SWEBoK and 7 other standards
- **Goal Model:** Shows stakeholder dependencies with —(D)—
- **Personas:** Bio, 3 ratings, typical activities, drivers, goals, pain points. Useful when real users are not available or there are too many to interview them all.
- **Domain Engineering:** Elicit requirements and scope of SPL. Determine commonalities and variabilities in terms of features
- **Application Engineering:** Bridge between customers and engineering teams
- **KPI:** Modelled with target, threshold and worst values for the conversion function (or use qualitative mapping)
- **Softgoals:** represent stakeholder concerns
- **Use Cases:** Title, ID, Actors, Intention, Preconditions, Main Alternatives, Postcondition(s)

- <<include>>: express commonality between different use cases
- <<extend>>: make optional interactions explicit or handle exceptional cases
- User Story: "As a (role), I want (thing) so that (benefit)."
- INVEST: Independent, Negotiable, Valuable, Estimable, Small, Testable (User Stories)
- Misuse Cases: Threatens and Mitigates
- Quality (NFR) Requirements:
  - Performance Speed
    - Execution speed
    - Response time, throughput
  - Robustness
    - Tolerates invalid input
    - Fail-safe/secure
    - Degrades gracefully under stress
  - Security
    - Controlled access to system/data
    - Protection against theft
  - Reliability/Availability
    - Fault-tolerant
    - Mean-time to failure
    - Data backups
  - Adaptability
    - Ease of adding new functionality
    - Reusable in other environments
    - Self-optimizing/healing
  - Usability
  - Scalability
  - Efficiency
  - Accuracy/precision
- Context Diagram: Data flow only. System in the middle, actors/things around it
- Waiting Place:  ,  Timer: 
- Three Cs of User Stories: Card, Conversation, Confirmation
- Requirements Management:
  - Manage changes to agreed requirements
  - Manage changes to baseline (increments)
  - Keep project plans synchronized with requirements
  - Control versions of individual reqs. and reqs. docs
  - Track requirement status
  - Manage relationships between requirements
- Requirements ID: Numbering based on doc section, dynamic renumbering, DB record identification, and symbolic identification (e.g. SEC1, SEC2)
- Requirement Attributes: Creation date, last update, author, stakeholders, version #, status, priority, stability, confidence, risk, rationale, comments, AC, test result, verification method
- Triage: Requirements prioritization
- Triage Techniques: Priority ranking, 100 dollar test, prioritization scales, Kano survey, Weiger prioritization, AHP comparison (1-9)

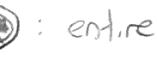
- Kano survey: Basic, performance, excitement, indifferent, reverse (functional vs. disfunc.)
- Weiger's Prioritization: $\frac{(\text{value \%})}{(\text{cost \%} \cdot \text{cost weight}) + (\text{risk \%} \cdot \text{risk weight})}$
- Verification: Are we building the product right?
- Validation: Are we building the right product?
- Requirements Analysis: Works with raw, incomplete requirements as elicited from stakeholders
- Requirements V&V: Works with negotiated and agreed upon domain requirements
- V&V Techniques: Simple checks, prototyping, functional test design, formal V&V, reviews & inspections
- Formal V&V: Simulations, testing, completeness/consistency testing, refinement checking, model checking, theorem proving
- Typical Requirements Issues:
  - Requirement not clear, needs clarification
  - Missing information
  - Conflict w/ other requirements
  - Unrealistic
- Anatomy of a Good Requirement:
  - Defines system under discussion
  - Defines a positive end result
  - Seek out system, end result, and success measure in every requirement
  - Write what, not how (i.e. no implementation)
  - Verb w/ correct identifier (shall or may)
  - Quality criteria
  - Feasible, Needed, Testable
- Scope: defined with a context diagram
- Goal: an objective or concern that guides the RE process, conveys intention
  - Needs to be verifiable to become a requirement
- Interview Objectives: Record information (to be used as input for requirements analysis), discover information about interviewee, reassure interviewee
- Brainstorming Roles: Scribe, Moderator/Leader
- KPI Examples:
  - Average Work Time (mins)
  - Monthly Infrastructure Cost ($)
  - Average Travel Time (mins)
  - Average Ongoing Cost ($)
- KPI Qualitative Mapping: More flexible ways of converting real world values to satisfaction values
- Qualities of a Good Model: Abstract, Understandable, Accurate, Predictive, Inexpensive

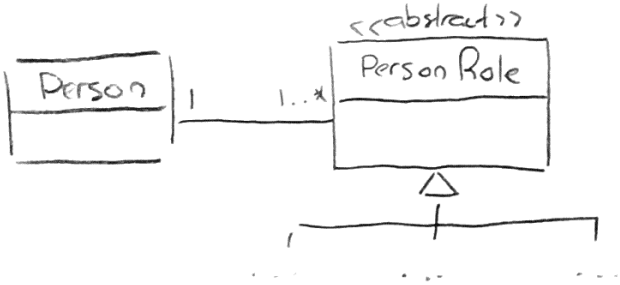
- Activity Diagram:  : entire activity,  : current flow only
- AD Partitions: Swimlane, Hierarchical Swimlane, Multidimensional Hierarchical Swimlane
- Player - Role Pattern:



- Abstraction - Occurrence Pattern:

- Context diagram: interacting systems and interacting users
- Systems Engineering Sandwich: Modelling + Requirements
- Feature Models: Variability, Commonality, Configurability
- Benefits of SPLs:
  - Improve product reliability
  - Improve usability
  - Shorten time-to-market
  - Improve consistency across products
  - Reduce production costs
- UML vs. UCM: Activity diagrams have data flow, use cases have dynamic stubs
- Availability: $\frac{MTBF}{MTBF + MTTR}$
- Project Intelligence: Based on data produced + decisions made
- Traceability Challenges: Accessing + Integrating Data, knowing what you want, connecting artifacts, querying the data
- Traceability Planning: Definition of links + attributes, stakeholders, needs, process, tools, queries, representations
- Mistakes: Failure to plan, ill-defined trace links, redundant trace paths, lack of unique project-wide IDs, missing trace links, traceability as an afterthought
- Baseline: Non-modifiable version of a document, enables comparison + management
- Suspect Links: Indicates that an element may have been affected by a change

- Activity Diagram:  : entire activity,  : current flow only
- AD Partitions: Swimlane, Hierarchical Swimlane, Multidimensional Hierarchical Swimlane
- Player - Role Pattern:



Abstraction - Occurrence Pattern:

- Context diagram: interacting systems and interacting users
- Systems Engineering Sandwich: Modelling + Requirements
- Feature Models: Variability, Commonality, Configurability
- Benefits of SPLs:
  - Improve product reliability
  - Improve usability
  - Shorten time-to-market
  - Improve consistency across products
  - Reduce production costs
- UML vs. UCM: Activity diagrams have data flow, use cases have dynamic stubs
- Availability: $\frac{MTBF}{MTBF + MTTR}$
- Project Intelligence: Based on data produced + decisions made
- Traceability Challenges: Accessing + Integrating Data, knowing what you want, connecting artifacts, querying the data
- Traceability Planning: Definition of links + attributes, stakeholders, needs, process, tools, queries, representations
- Mistakes: Failure to plan, ill-defined trace links, redundant trace paths, lack of unique project-wide IDs, missing trace links, traceability as an afterthought
- Baseline: Non-modifiable version of a document, enables comparison + management
- Suspect Links: Indicates that an element may have been affected by a change