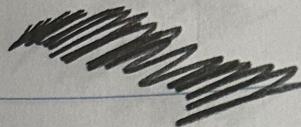


COE691 - Notes



Software: Collection of executable programming code

- Implies large, robust, reusable, evolving
- Teaches you how to design

Software Engineering: Application of systematic, disciplined, quantifiable approach to development, operation, & maintenance of software

- 3 basic forces:

- Cost: Cost to develop & deliver project
- Schedule: Turn-over speed of project
- Productivity: Models above two via O/I resources
- Software Quality: ISO Standard (6 attributes - RUMPEF)

- Reliability
- Usability
- Maintainability
- Portability
- Efficiency
- Functionality

Software Problem

Software Development Process (Layered Approach)

- Software Development (1st Layer)
 - Requirements: Output is Software Requirements Spec (SRS) doc
- Software Design (2nd layer)
 - Design: Output is architecture, description, or logic design doc
- Programming (3rd Layer)
 - Code/Implementation: Output is Code
 - Testing: Output is testing log
- Software Design (2nd layer)
 - Maintenance: Output is updates (Also delivery)
- Error. Dist.

- 10-20%: Req, S Design / 30-50%: Testing

- 20-30%: Coding

Process Models: Abstract representation of projects

- Traditional:

- Waterfall:

- Oldest + Most Used

- Layered approach, next phase starts when prev is finished

- 6 Stages: Req, Design, Code, Test, Deliver, Maintenance

- Prototyping:

- Used for risky projects

- 4 Stages: Req (often 3), Design, Code, Test

- Only proto features needing more clarity

- Iterative:

- Used in product dev

- 3 Stages: Design, Code, Analyze / Test

- Repeat stages n # of times

- Timeboxing:

- Iterative 2.0

- 3 Stages: Req, Build, Code

- Repeat, each stage must be done in timeframe

- Linear: T, 2T, 3T / Pipeline: T, T+T/3, T+2T/3

- Modern:

- Agile:

- Used for mobile apps

- 5 Stages: Brainstorm, Design, Code, Test, Deliver

- Cycle model, work in short incr

Requirement Eng: Process of collecting, understanding, & evaluating reqs

- Inception: Start, generate ideas

- Elicitation: Reqs decided via stakeholders

- Analysis & Negotiation: Analyze + Fix conflicts

- Specification: Create req doc (70% of defects)

- Validation: Confirm / Finalize } 30% of defects

- Management: Update

} MAEVIS

Business Reqs: Focus on company reqs (\uparrow Market Share, \downarrow HR Costs)

User Reqs: Focus on customer reqs (Make an appointment)

Software Reqs: Focus on FA & NFR

- FA: What program must do
- NFR: Quantifiable measure of program
 - Safety: Absence of disasters
 - Usability: Ease of use
 - Reliability: Complete tasks as outlined
 - Robustness: Handle errors & input stress
 - Dependability: Can place reliability
- Function \rightarrow Req \rightarrow Task \rightarrow NFR

} SURAD

Sys Dev:

- 2 Approaches
 - Structured: Focuses on models (Data-flow, ERD)
 - Obj-Oriented: Treat as collection of objects (Class, Comp, Seq.)

RE Lifecycle:

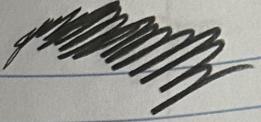
- Elicitation: Identify reqs
- Analysis: Understand & analyze reqs
- Specification: Document reqs
- Evaluation: Ensure reqs

} SAFe

Inception:

- Activity: (Define scope, perform feasibility study, identify stakeholders)
- Artifacts: (Business Case (Doc that outlines business objs), Use case model, risk list)
- Preliminary Investigation: Evaluate opportunities
- Fact-Finding Techs: Gather data using questionnaires
- Evaluating Feasibility: Analyze proposed system via
 - operational, technical, economic, organizational, & schedule
- SWOT Analysis: Strengths, Weaknesses, Opportunities, Threats

- Present results



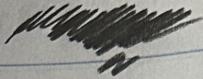
Elicitation:

- Understand app domain (Understand business, app domain, specific problem, needs & constraints of stakeholders, acquisition & proj management, reqs & sys engineering, technologies being involved)
- Identify source of reqs
- Analyze stakeholders
- Select techs, approaches, & tools
- Elicit reqs
- Stakeholders:
 - Primary (Direct affect) vs Secondary (Indirect affect) P = Power
 - HI & HP (Manage Closely), HI & LP (Keep informed), LI & HP (Keep satisfied), LI & LP (Monitor)
 - 3 primary groups (Users, Clients, Tech Staff)
 - 5 engagement levels (Unaware, Assistant, Neutral, Supportive, Leading)

* I = Interest

Elicitation Techs:

- Important for obtaining critical info from Subject Matter Experts (SME) & stakeholders
- 2 categories:
 - Classic:
 - Includes interviews, introspection, questionnaires, analysis of existing sys, doc analysis, observation & social analysis
 - Introspection: Imagine sys required
 - AOES: Examines current sys, product, or process
 - Doc Analysis: Review business plans, software docs, regulatory docs
 - OSEA: Passive, Active, Exploratory, & Ethnography
 - Modern:
 - Includes scenarios, brainstorming, prototyping, joint/rapid app dev, task analysis, workshop, reverse eng
 - Task Analysis: Break task into small parts
 - JAD: Who, What, When, Where, Why, How
 - Prototyping: Horizontal, Vertical, Throw-away, High vs Low fidelity



Classification & Organization:

- FR Classification
 - Name of feature
 - Desc & Priority
 - Stimulus / Response sequences
 - List of associated FR
- NFR Classification (FURPS)
 - Functionality
 - Usability
 - Reliability: MTBF (Mean time between failure), MTBSI (MTB sys/service incidents), MTTR (Mean time to repair)
 - Performance
 - Supportability

Prioritization:

- Analytical Hierarchy Process (AHP):
 - Cost-Value Diagram: Used to plot priority
 - Set up n # reqs \rightarrow Perform pairwise comparisons $\left(\frac{n(n-1)}{2}\right) \rightarrow$ Use avg over normalized columns \rightarrow assign each req relative value
 - Perform 2x, 1 for relative value, 1 for estimate relative cost

Negotiations & Stakeholder Conflicts:

- Severity of Conflict:
 - Mutually Exclusive (Impossible)
 - Non-Interfering (Can be combined)
 - Interfering (Negotiate)
 - Inclusive (One involves another)
- Conflict Resolution:
 - Negotiate (Work together)
 - Competition (Prioritize personal gain)
 - 3rd-Party (Outside adjudicator)
- Bidding (Bargain)

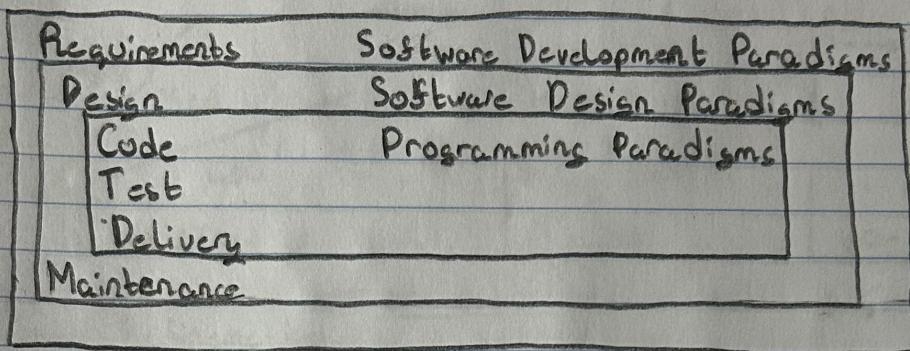
Goal Analysis & Modeling:

- Goal model is a hierarchy of goals, relating high-level goals to low-level sys
- 3 models: i*, KAOS (Knowledge Acquisition in Automated Spec), UML Use Case
- i* (I Star)
 - 2 types of diagrams:
 - Strategic Dependency (SD): Defines relationships between roles
 - Strategic Rationale (SR): Analyze goals from SD
 - Goal-Oriented Requirement Language
 - Intentional Elements:
 - Goal: Condition/Situation to achieve
 - Softgoal: NFR
 - Task: How to accomplish goal
 - Resource: Object available to use
 - Belief: Represent assumptions/conditions
 - Indicator: Satisfaction Level
 - Intentional Relationships:
 - Contribution: Describable influence (\longrightarrow)
 - Dependency: Interdependencies ($\neg\rightarrow$)
 - Decomposition: Sub-Components (\perp)
 - Correlation: Side-effect (\dashrightarrow)
 - Means-End: How to achieve goal ($\rightarrow\!\!\!D$)

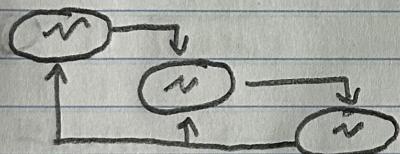
* Contribution Types	\perp - Make / \perp - Help \perp - Some Pos / \perp - Unknown Some Neg / \perp - Breaks \perp - Hurt
+ - Make / \perp - Help	
+ - Some Pos / \perp - Unknown	
- Some Neg / \perp - Breaks	
\perp - Hurt	
 - Actor:
 - Active object carrying out actions
 - Goal-Model Analysis
 - Qualitative: Use contribution types, importance (High, Med, Low) & satisfaction levels
 - Quantitative: Contribution & Satisfaction [-100, 100], Importance [0, 100]
 - Hybrid: Mix of prev 2

COE691 - Diagrams

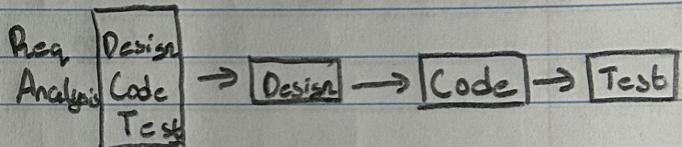
SD Process:



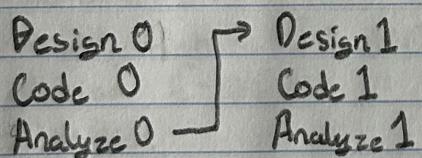
Waterfall Model:



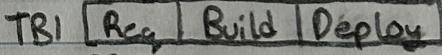
Prototyping:



Iterative Model:

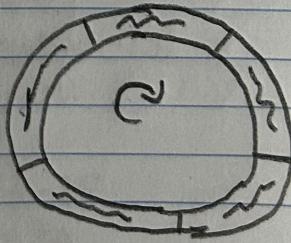


Timeboxing:

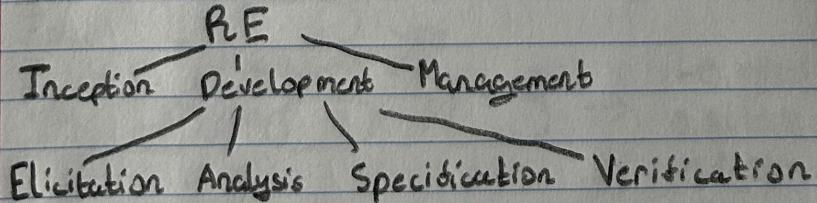


TB2 "

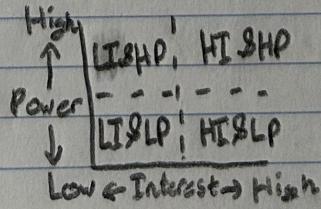
Agile Model:



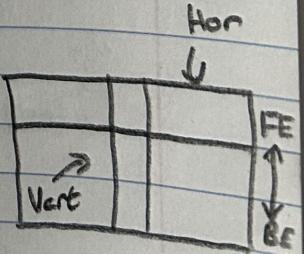
RE:



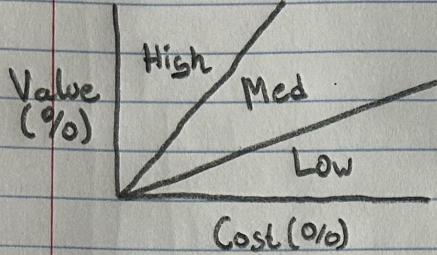
Power-Interest:



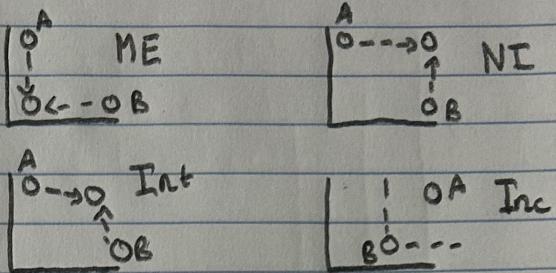
Prototyping:



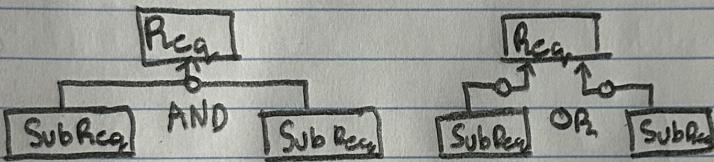
Cost-Value:



Severity of Conflict:



Goal Refinement:



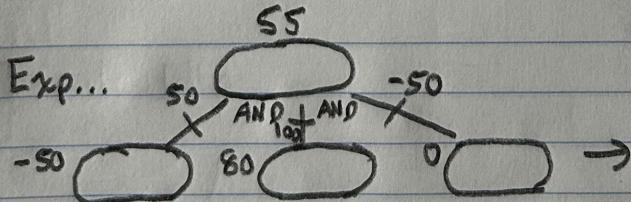
+ : Help / ++ : Make
- : Hurt / -- : Break

Goal-Modelling (Cont...):

- ✓ : Satisfied [100]
- ✗ : Weakly-Satisfied [0, 100]
- ? : Unknown [0]
- ✗ : Weakly-Denied [0, -100]
- ✗ : Denied [-100]
- ≥ : Conflict
- : None

Hybrid: Contribution:

- Make (100)
- Some Positive (75)
- Help (25)
- Unknown (0)
- Hurt (-25)
- Some Negative (-75)
- Break (-100)

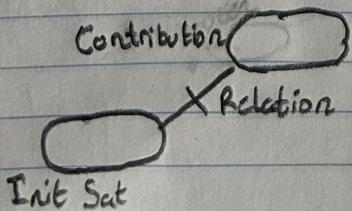


$$-0.5(50) = -25$$

$$0.8(100) = 80$$

$$0(-50) = +0$$

$$\underline{55}$$



* [Init Sat * 0.01(Contribution)]