

OMSE 532 - Week 8

Program Families

Architectures for Sets of Systems

© Stuart Faulk 2006

1

Outline

- What is a program family?
- Why develop programs as families?
- The role of architecture
- Designing family architectures

© Stuart Faulk 2006

2

Program Families

We consider a set of programs a family if they have so much in common that it pays to look at their common aspects before looking at the aspects that differentiate them.

- David L. Parnas

- What are some examples of program families?

© Stuart Faulk 2006

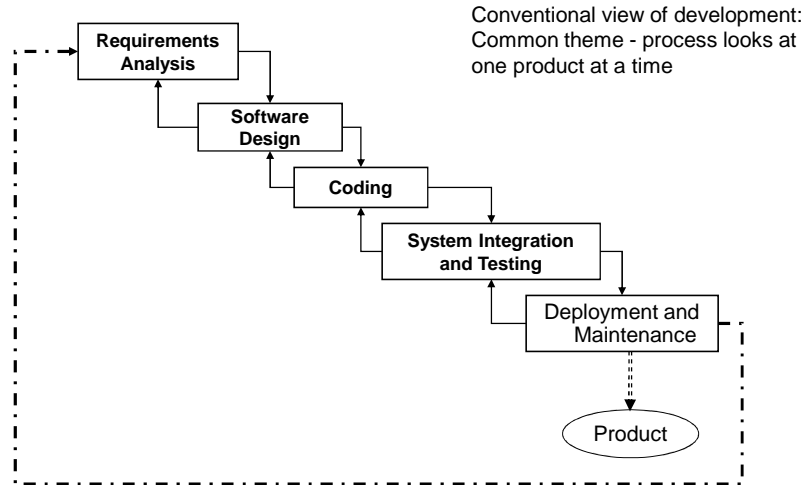
3

Why Develop Programs as Families?

© Stuart Faulk 2006

4

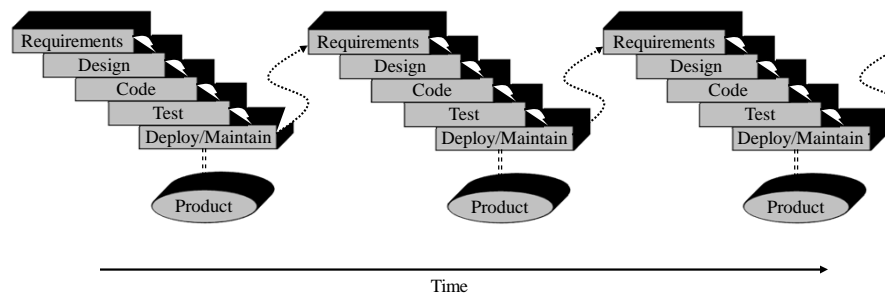
Merry-Go-Round of Sequential Development



© Stuart Faulk 2006

5

Sequential Development Over Time



... a result of "tactical software engineering"

© Stuart Faulk 2008

6

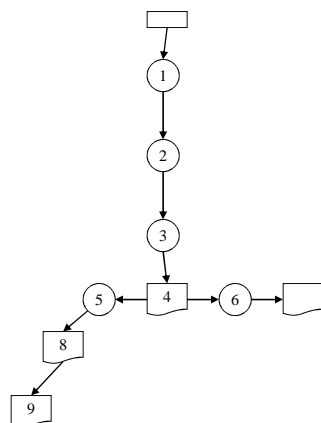
Inefficiencies of Sequential Development

- Hypothesis: much of software development is re-development.
 - Software inevitably exists in many versions
 - Seldom develop truly new applications
- Implication: typically much in common among our systems
...But very little is reused
 - Difficult to identify commonalities and differences
 - Difficult to reuse code components
 - Difficult to add desired feature to existing design
 - Difficult to adapt other work products (if they exist)
 - Generally easier to re-do than re-use
- What makes work products difficult to reuse?

© Stuart Faulk 2006

7

Sequential Development



- Incomplete program
- Working program
- Design Decision

- System developed through sequence of design decisions leading to product (8)
- Developing new product version requires backing up
 - Some decisions won't apply to new version
 - Must back up to point where decisions can be re-made. (5 or 6)
- How far we need to back up depends (roughly) on the order of decisions

From Parnas: On the Design and Development of Program Families

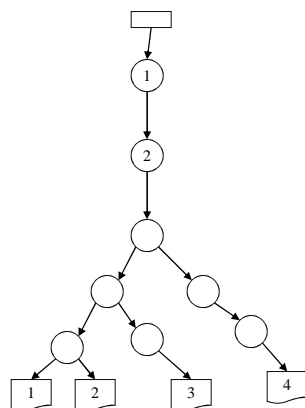
8

Objectives of Family Development

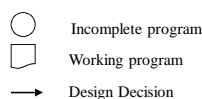
- Exploit commonalty when building similar systems
 - Deploy systems in multiple versions
 - In maintenance, quickly build new versions of a system
 - Produce deliverable code and documentation rapidly
 - Reduce cost, improve quality
- Design products to be reused (changed, extended)
 - Easy to add new features/capabilities
 - Easy to produce different versions
- Focus on reuse *conceptual structures*
 - Code is not the hard part, low ROI
 - Want to reuse requirements, design, etc.
- Requires a strategic view of development (encompasses multiple developments over time)

9

Family Development Model



- Development of new system begins from an intermediate stage
 - Order of decisions is critical
 - Intermediate representation is important
- Branching = different decision
 - All decision above a branch are in common to the family members
- Most similarity achieved by making as many common decisions as possible before creating differences
- What would be the role of architecture?



From Parnas: *On the Design and Development of Program Families*

© Stuart Faulk 2006

10

Role of Architecture

- Architecture provides the basis for making and representing common design decisions for a family
 - Use architecture to instantiate common design decisions
 - Architectural representation provides the “intermediate representation” for the family
 - Instances of a family share common architecture, differ in design details
- How do we systematically construct one?
 - Developing software systems as a family
 - Software product lines (next week)

© Stuart Faulk 2006

11

Family Architecture Design

© Stuart Faulk 2006

12

Architectural Design Process

Design process: makes decisions & tradeoffs to optimize the mutual satisfaction of quality requirements

1. Creating the business case for the system
2. Identify & prioritize behavioral and quality attributes
3. Design the architectures
4. Representing and communicating the architecture
5. Analyzing or evaluating the architecture
6. Implementing the system based on the architecture
7. Ensuring the implementation conforms to the architecture

© Stuart Faulk 2006

13

Family Design Process Differences

1. Creating the business case for the system
 - How does the business case for a family differ?
 - What goals are added?
2. Identify & prioritize behavioral and quality requirements
 - What additional information is needed for a family?
 - Where does it come from?
3. Design the architectures
 - How do we use that information in the design process to create a common architecture?
4. Representing and communicating the architecture
 - What is the product of the family design process?
5. Analyzing or evaluating the architecture
 - What properties would we analyze for?
6. Implementing the system based on the architecture
 - How does implementation of a family member differ?

© Stuart Faulk 2006

14

Inputs Needed

1. Creating the business case for the system

- How does the business case for a family differ?
 - What are the cost/benefit considerations?
- What goals are added to single family development?

2. Capture behavioral and quality requirements

- Assume that we are gathering behavioral and quality requirements
- What additional information do we need about requirements?
 - Ex. Assume the family has just two members to start and a small set of differences
- Where would the necessary information come from?
- How would we capture and communicate it?

Inputs Needed

1. Creating the business case for the system

- Business case looks at developing a family of products over time
 - What are the cost/benefit considerations?
- Goals include developing common, reusable assets

2. Capture behavioral and quality requirements

- Identify the behavioral and quality requirements that are common to members of the family
- Identify requirements that differ from one member to the next
- Capture in requirements specifications

Commonalty Analysis

- A process for characterizing members of a family
- Objectives
 - Terminology: identify & define common domain terminology
 - Identify commonalties: identify and specify requirements family members have in common (common assumptions)
 - Identify variabilities: identify and specify how family members can vary
 - Describe kinds of variation
 - Specify ranges of variation
 - Identify issues: problems to be resolved
- Participants
 - Domain experts: Provide domain expertise
 - Moderator: Directs and guides analysis
 - Recorder: Produces analysis
 - Other stakeholders(?), e.g. customer representatives

© Stuart Faulk 2006

17

Family Design Steps

3. Design the architectures
 - What are the additional design goals?
 - How do we use information about commonalties and variabilities?
4. Representing and communicating the architecture
 - What is the product of the family design process?
 - Hint: how would we use the product in the development of a new family member?
5. Analyzing or evaluating the architecture
 - What properties would we analyze for?

© Stuart Faulk 2006

18

Family Design Steps

3. Design the architectures

- Use information about commonalities and variabilities to guide design of a common family architecture
- Design common aspects for reuse, make decisions about common requirements first
- Design parts that will vary for modifiability, bind these decisions late

4. Representing and communicating the architecture

- Specify the common architecture
- Specify how to implement variation

5. Analyzing or evaluating the architecture

© Stuart Faulk 2006

19

Family Design Approach

- Conceptual approach using familiar concepts (there are others)
- Approach based on “design for change”
 - Most-solid-first: apply to common elements
 - Information hiding: apply to variations
- How is the system decomposed into parts?
 - System is decomposed into a hierarchy of information-hiding modules.
 - Structural decisions common to the family members are made first – result in overall structure and interfaces
 - Variabilities are pushed to the leaf modules and hidden when possible

© Stuart Faulk 2006

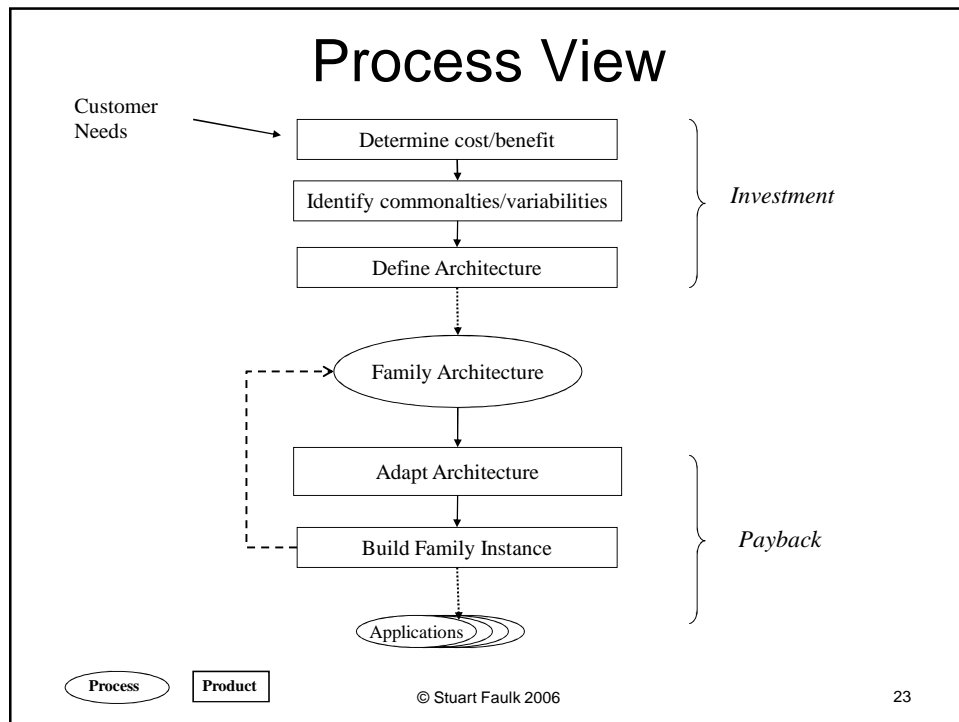
20

Evaluation and Outputs

- 5) Analyzing and evaluating the architecture
 - What qualities or properties would we analyze the design for?
 - What information would we use?

Evaluation and Outputs

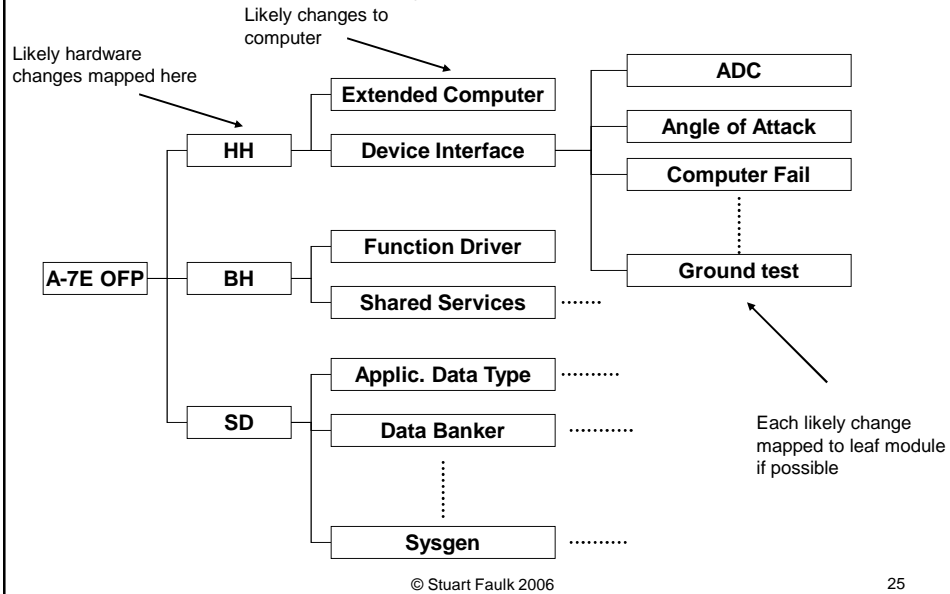
- 5) Analyzing and evaluating the architecture
 - Analyze if the common architecture
 - is reusable for all identified members of the family
 - Is easy to change (modify) to accommodate the anticipated variations
 - Use the outputs of the commonality analysis (need not be formal)



Examples of Families

- Have seen several examples of family architectures
- A-7E software
- Flight simulation
- Spam Filter (depending on approach)

A Family Architecture



Assertion

- It always pays to build a systems as a family
- Hint: what is the relationship between designing for a family and designing for maintainability?

Summary

- When developing similar systems pays to develop as a family
- Applies to variation produced by maintenance
- Approach: develop a family architecture
 - Analyze what's in common and what varies
 - Apply most solid first
 - Put decisions that won't change in the architecture
 - Hide decisions that will vary

27

Questions?

© Stuart Faulk 2006

28