ECEN 5053-002

Developing the Industrial Internet of Things

Week 7 - Lecture

Project planning, Staffing, Execution, BOM

Dave Sluiter - Spring 2018

Be Boulder.

University of Colorado Boulder

Material

- Product Development Process
- Documentation
- Building a Bill-Of-Materials (BOM)
- Product teardown analysis

Learning Outcomes

- Students will learn about:
  - A Product Development Process
    - The importance of defining a process
  - The importance and role that documentation plays in the engineering workplace
  - How to build a Bill-Of-Materials (BOM)

A Process?

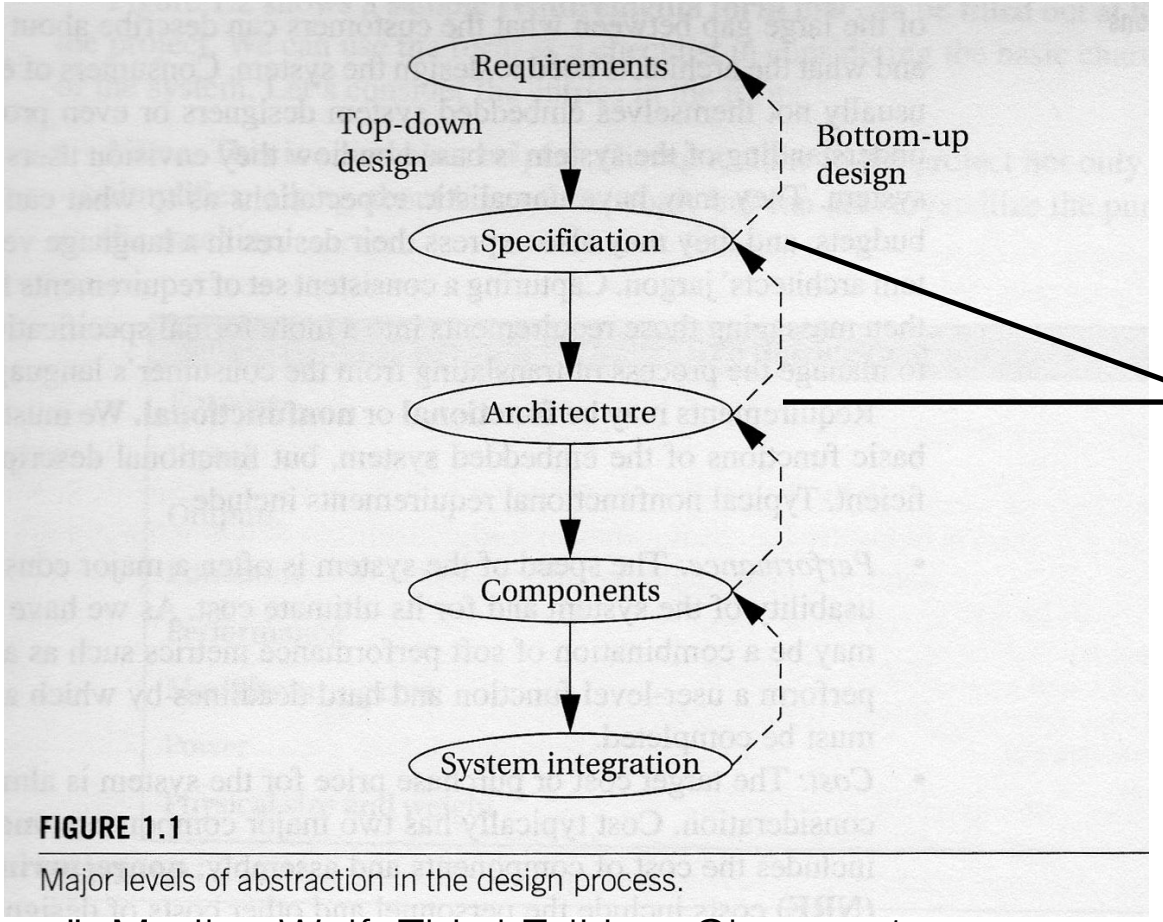


FIGURE 1.1

Major levels of abstraction in the design process.

I combine these

From “Computers as Components”, Marilyn Wolf, Third Edition, Chapter 1

A Process?

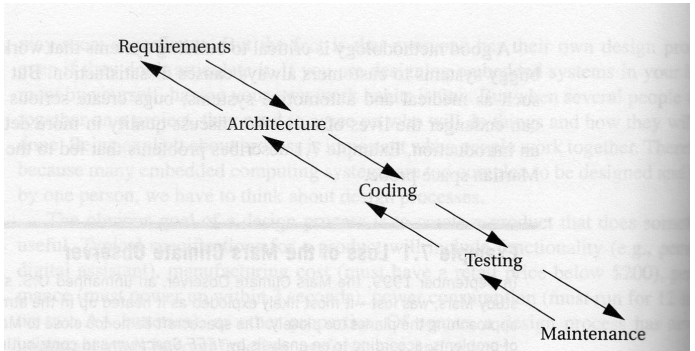


FIGURE 7.1
The waterfall model of software development.

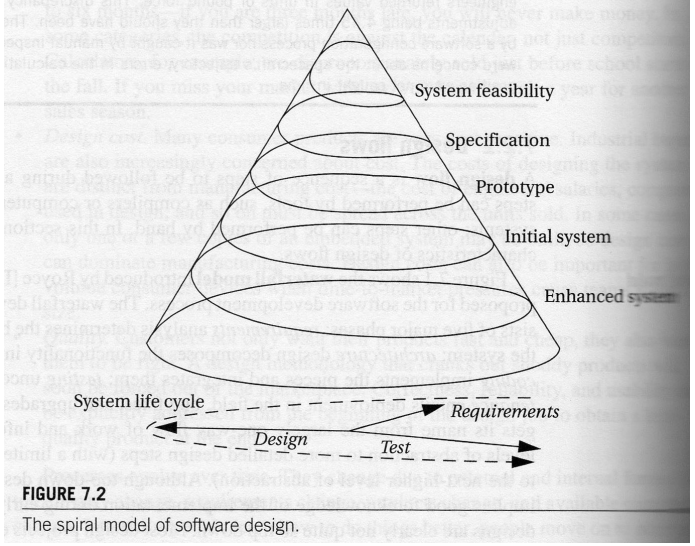


FIGURE 7.2
The spiral model of software design.

From “Computers as Components”, Marilyn Wolf, Third Edition, Chapter 7

A Process?

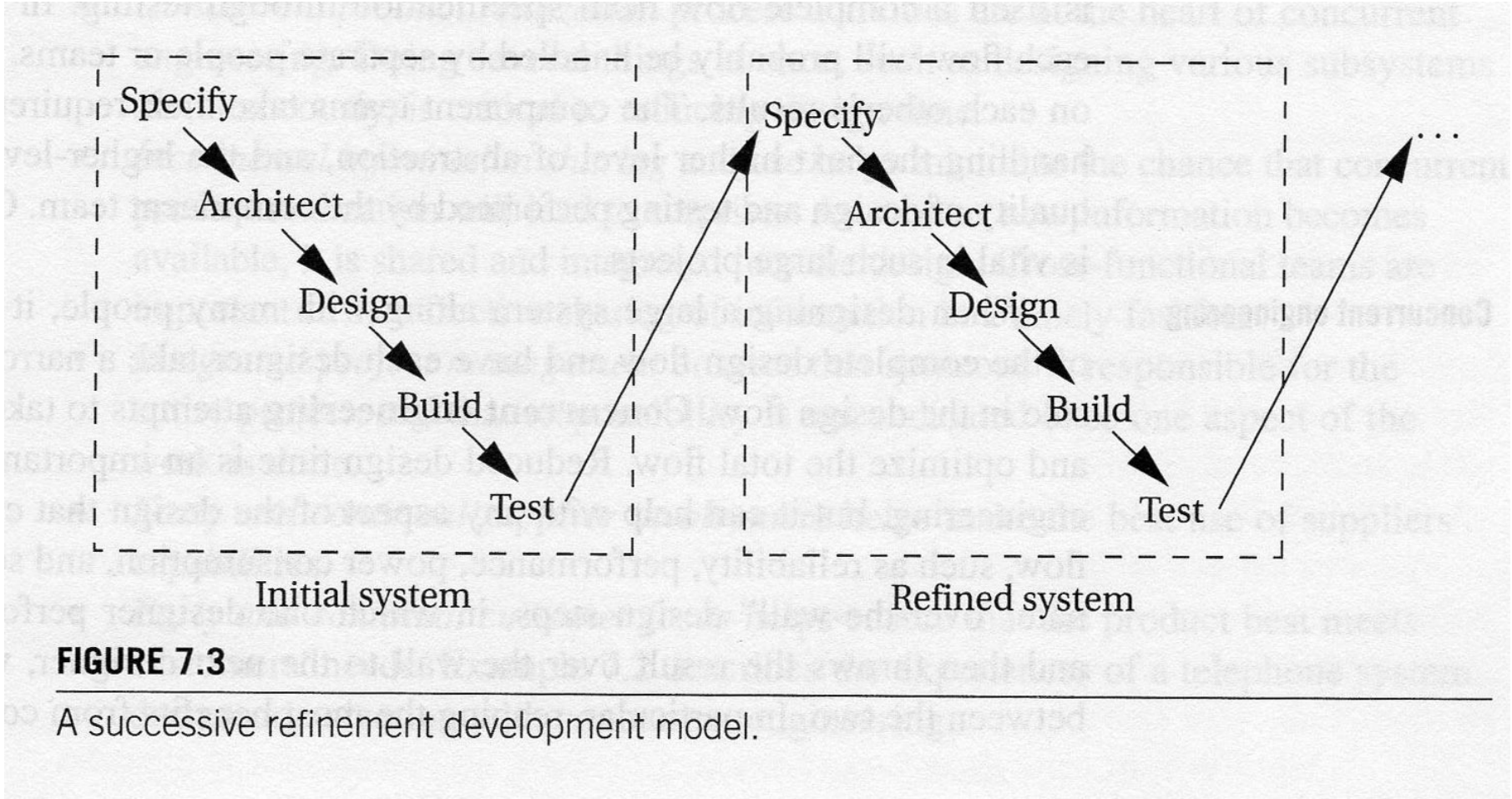


FIGURE 7.3

A successive refinement development model.

From “Computers as Components”, Marilyn Wolf, Third Edition, Chapter 7

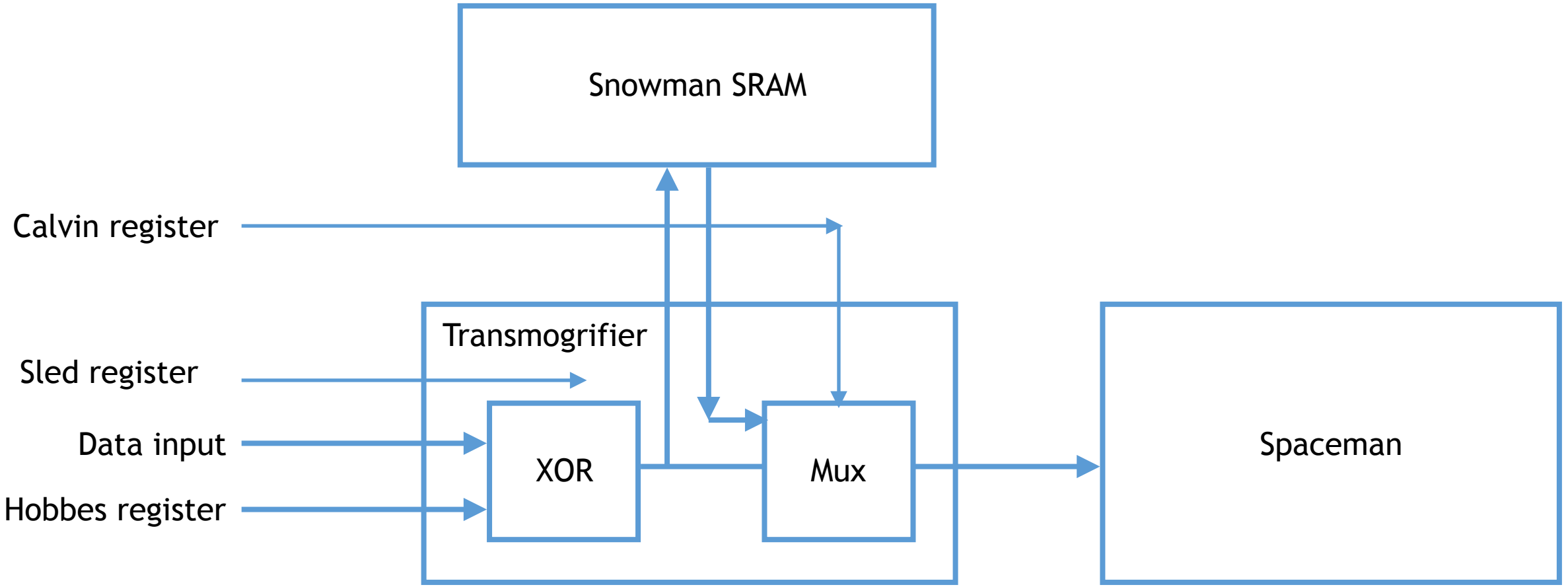
Having a Process is Good

- Requirements Document - *Always start with requirements!*
  - *"**shall**" statements - the product shall do blah blah blah…*
- Architecture Specification - Write down "**what**" and "**how**", but not a detailed implementation. Many engineers have trouble with this abstraction.
- Include as many figures as time allows
  - Figures (drawings) remove the ambiguity of words

Figure Example

- Which is more clear?
- Data shall enter the transmogrifier where is it is XOR’ed with the Hobbes register value. The data can then be temporarily stored in the snowman SRAM, or forwarded on to the Spaceman module as per the Sled register value and Calvin register settings.

Figure Example



Having a Process is Good

- Components - Design and build all the pieces
- System Integration - Putting it all together
  - Loss of the Mars Climate Observer (imperial vs. metric units)
- But there is way more to it than these simple descriptions

Requirements

- Where do requirements come from?
  - Customers
  - Internal sources
    - Sales
    - Marketing (competitive analysis)
    - Management
    - Engineering (lessons learned from previous projects)

Contents

- Staffing
- Organization
- Roles and Responsibilities
- The Process

Staffing

★Staffing

- Organization
- Roles and Responsibilities
- The Process

Staffing

- It is in a company's best interest to staff a project appropriately
- But what is appropriately?
- The answer is:
  - It depends, as not all projects represent equivalent complexity, duration and resources (people).

Gauging Project Complexity

- Management looks at any given project as the sum of three vectors (metrics)
  - Scope
  - Schedule
  - Resources

Scope

- A measure of the design space, complexity
  - What is the feature list? How many features?
  - How much of the previous design can be leveraged?
  - How much new design work must be completed?

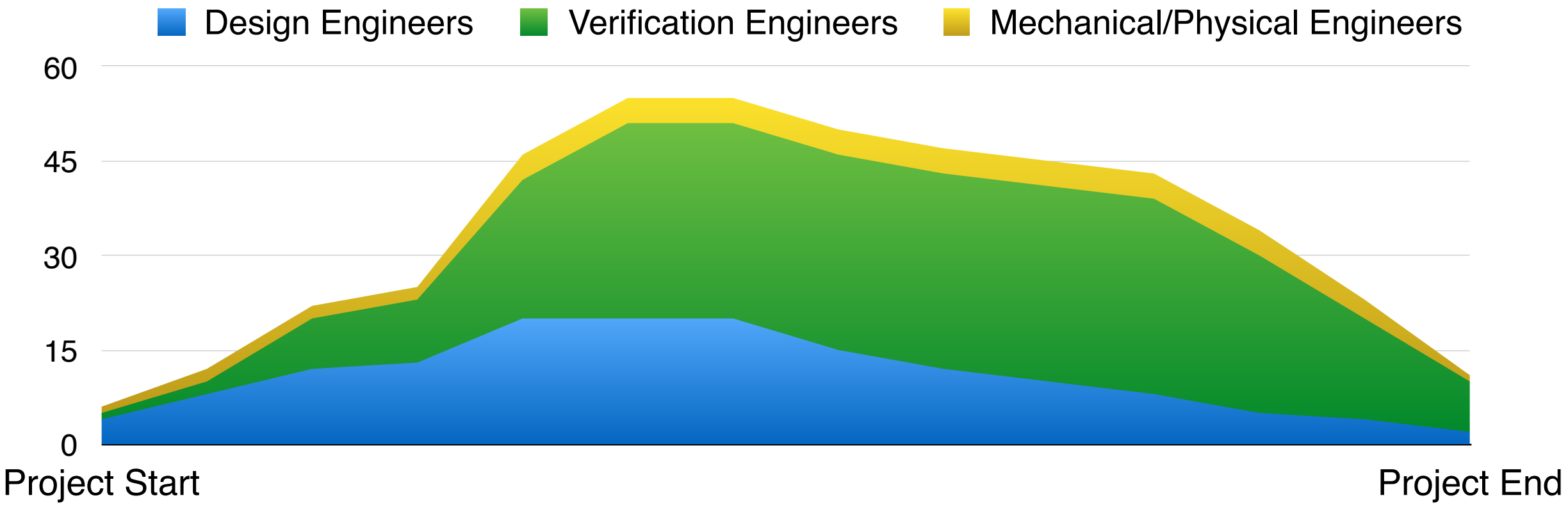
Schedule

- When does the project need to be completed?
  - According to management, sales and marketing?
  - According to customers?
    - When do customers want test units? (qualification units)
    - When do customers want the product in volume?
- These “wants” do not always agree

Resources

- Staffing
  - How many engineers are available?
    - Can we use the people we already have?
    - Is hiring required?
  - What does the heap chart look like?

Heap Chart (con't)

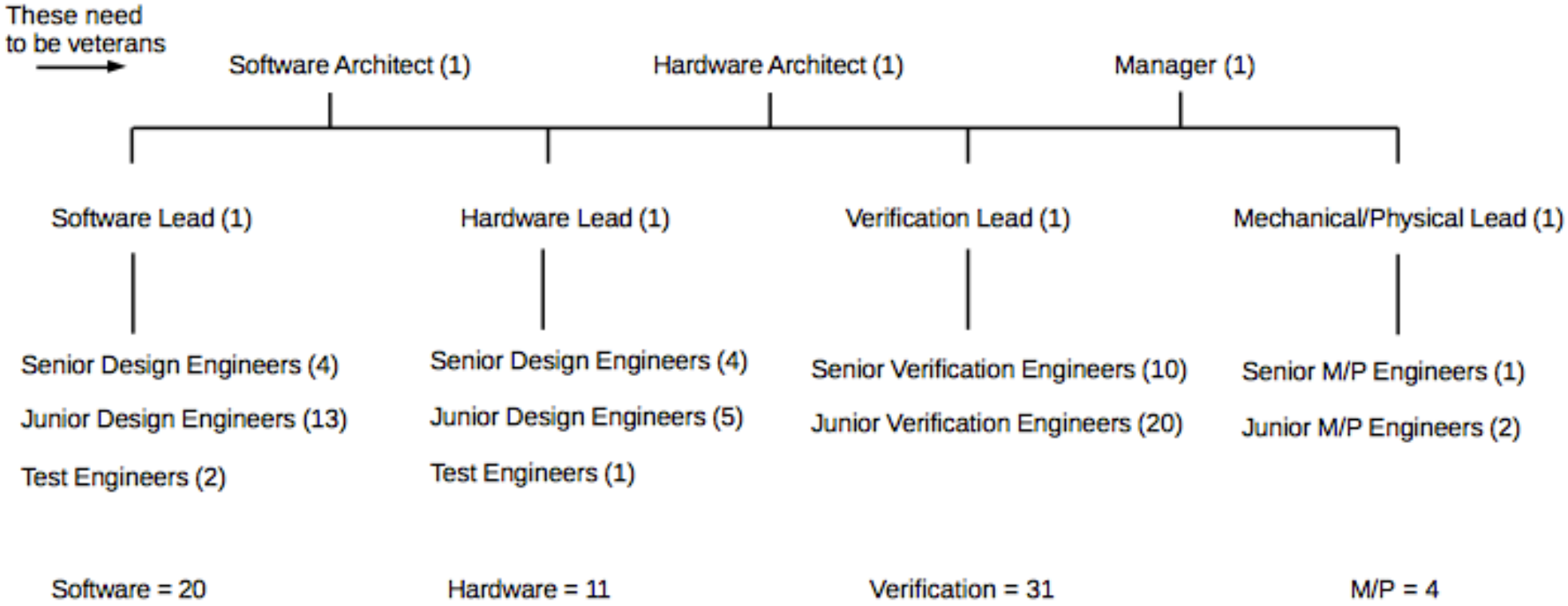


- The area under the curve represents **one** of the costs of doing business (OpEx)
- Management is highly motivated to minimize this expense

Organization

- Staffing
- ★Organization
- Roles and Responsibilities
- The Process

Organization (con't)



Note the ratio of hardware to software to verification engineers

Roles and Responsibilities

- Staffing
- Organization
- ★Roles and Responsibilities
- The Process

Roles and Responsibilities

- When employees are unclear on roles and responsibilities, mass chaos can ensue !

Source: Invasion of the Body Snatchers, 1956, Allied Artist Pictures

ASIC Design Example

Roles and Responsibilities (con't)

- Manager
  - Project schedule, task assignments, scheduling reviews, annual performance reviews, vacation planning
- Architect(s)
  - Requirements and architecture specification documents
  - **All technical decision authority "should" lie with these person(s)**
- Software/Hardware Lead(s)
  - Implementation trade-offs, low-level documentation: micro-architecture documentation, register definitions, programmers guide.
- Verification Lead
  - Defines testbench architecture and test approaches
  - Creates template document for module level test plans
- Mechanical/Physical Lead
  - Product mechanical design, thermal analysis, shock and vibration

The Process

- Staffing
- Organization
- Roles and Responsibilities

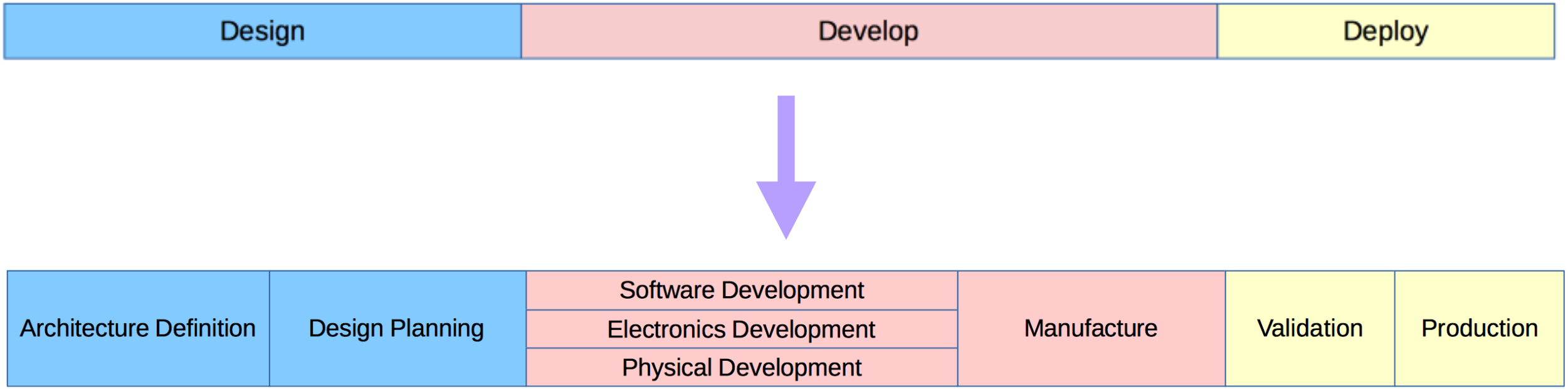
★The Process

Project Process

- Goals:
  - Minimize the area under the heap chart curve (head count, duration)
  - Produce highest quality product possible within the scope, schedule and resources, that meets the requirements.
- Jumping in and starting to code before the product is defined is a "*last-century*" design practice
- A modern design process uses a "3D" strategy that:
  - Designs first
  - Develops second
  - Deploys third

So what happens?

time



Design

The design is "done" here

Architecture Definition

Actions and Deliverables:

* Write requirements doc
  "shall" statements
  features, cost, power
  size, weight, performance
* Write architecture doc
  What it does, How it works,
  data and control flow, UML diagrams
  Division of labor between HW
  and SW
* Build schedule, **top-down**
* Performance models

Exit criteria

Reviews completed?

Design Planning

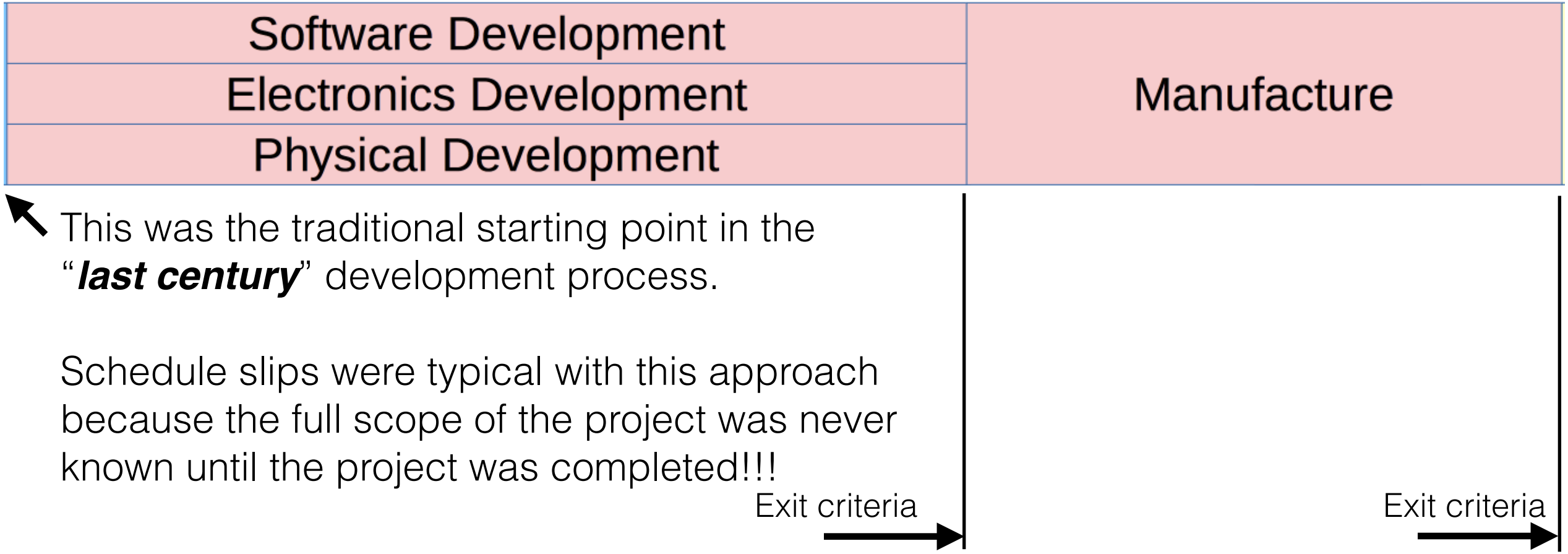
Actions and Deliverables:

* Initiate Request For Quotes from vendors, make vendor selections
* Write low-level specification docs:
  Hardware, data & control flow, state mach.
  Software, hardware interaction, errors
  Test plans - every module has a test plan
* Write testbench specification
  How will this design be verified?
* Build the "real" schedule, **bottom-up**
  We have real numbers now!

Exit criteria

Reviews completed?

Develop

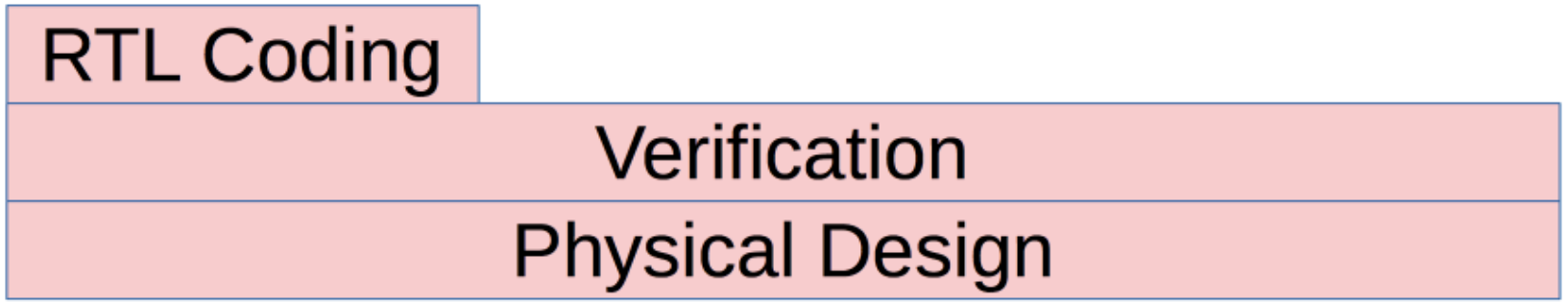


This was the traditional starting point in the "***last century***" development process.

Schedule slips were typical with this approach because the full scope of the project was never known until the project was completed!!!

Electronics Development

Assuming an Application Specific Integrated Circuit (**ASIC**)



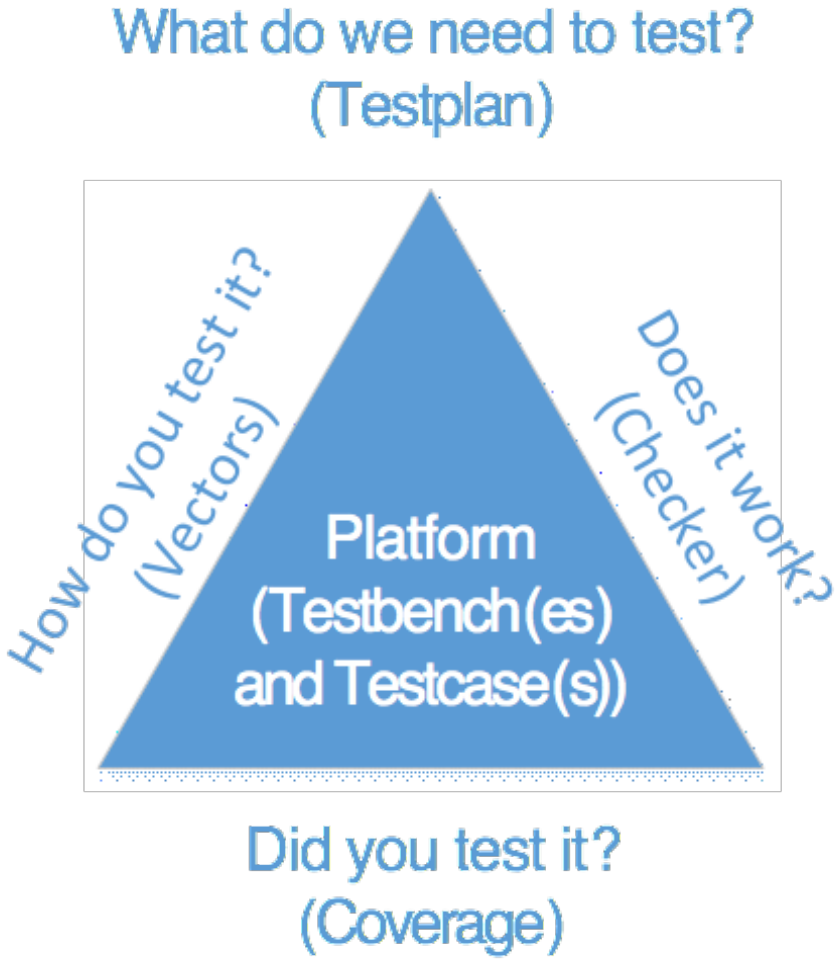
Actions and Deliverables:

* With the design completed, the RTL coding phase becomes very short. The RTL code literally just "falls out" because each engineer understands in detail what each module in the design must do.
* It then becomes a race between **Verification** and **ASIC physical design,** tasks like:
  - *Pin placement*
  - *Cell placement (flops and gates)*
  - *Clock tree insertion*
  - *Routing*
  - *Static timing analysis*
  - *Scan test insertion and ATPG*

Exit criteria

Reviews completed?

Verification Process



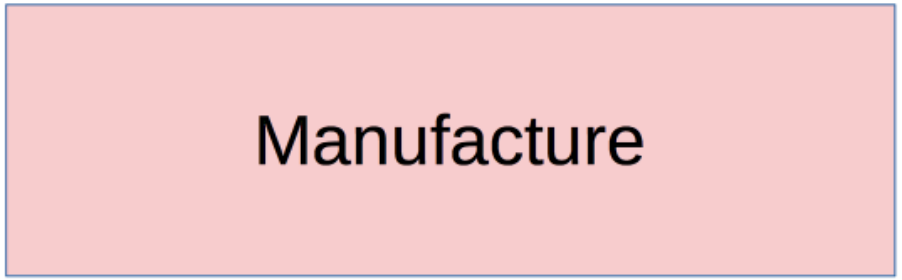
Verification Process

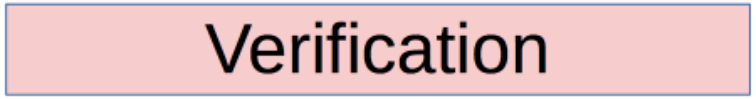
- Direct testing
  - Write a specific testcase and checker to test a specific function
- Constrained random verification
  - Write a random generator then measure that you tested a specific function
- How do you measure? How do you know you've simulated enough?
  - Designers write coverage assertions into the code
  - Coverage metrics are accumulated over many thousands of simulation runs
  - Results of these metrics tell you when you're done

ASIC/FPGA Verification Information

- Cadence developed OVM
- Synopsys developed VMM
- Merged into UVM
  - http://www.accellera.org/downloads/standards/uvm
- History: https://www.quora.com/What-exactly-is-the-difference-between-OVM-UVM-and-VMM-verification-methodologies-and-what-do-they-signify

Manufacture





“tape out”,
“tape release”
spend the
big $money!

* Wait for prototype ASIC parts to come back from fabrication.
* Not uncommon for verification to continue for some time, looking for additional bugs.

Exit criteria

receive prototype parts

Deploy

Validation	Production

Actions and Deliverables:

* Post silicon validation,
  Does the ASIC work? Does the Product work?
  Create list of bugs - errata
  Virtual test cases *- RTL Simulation, small # test cases*
  Veloce (*Mentor****) / Palladium*** *(Cadence) - Modest # test cases*
  FPGA *- Large # test cases*
* Product integration, build:
  Engineering Models
  Customer Test Units, aka Qualification Units
  Perform: *Thermal, shock and vibration testing*
* ***Validate Security*** *- Internal review + Hire an outside security firm to attempt to compromise your system*

Last exit criteria →

* Enter customer support phase

Validation

- What does it mean?
- Build in-form factor product
- Run functional tests to prove (Validate) that all features and requirements are met
  - May uncover bugs
- Burn-in
  - Run some number of devices for some number of hours
  - For example: 1000 devices for 1000 hours
  - Sometimes referred to as RTD (reliability demonstration testing)
- Run shock, vibration and thermal stress tests
  - Typical CMOS silicon operating junction temperatures (Tj) ~ 0 to 95C, may go up to 115C. No one I'm aware of allows Tj to reach 125C.

Validation

Thermal stressing a component

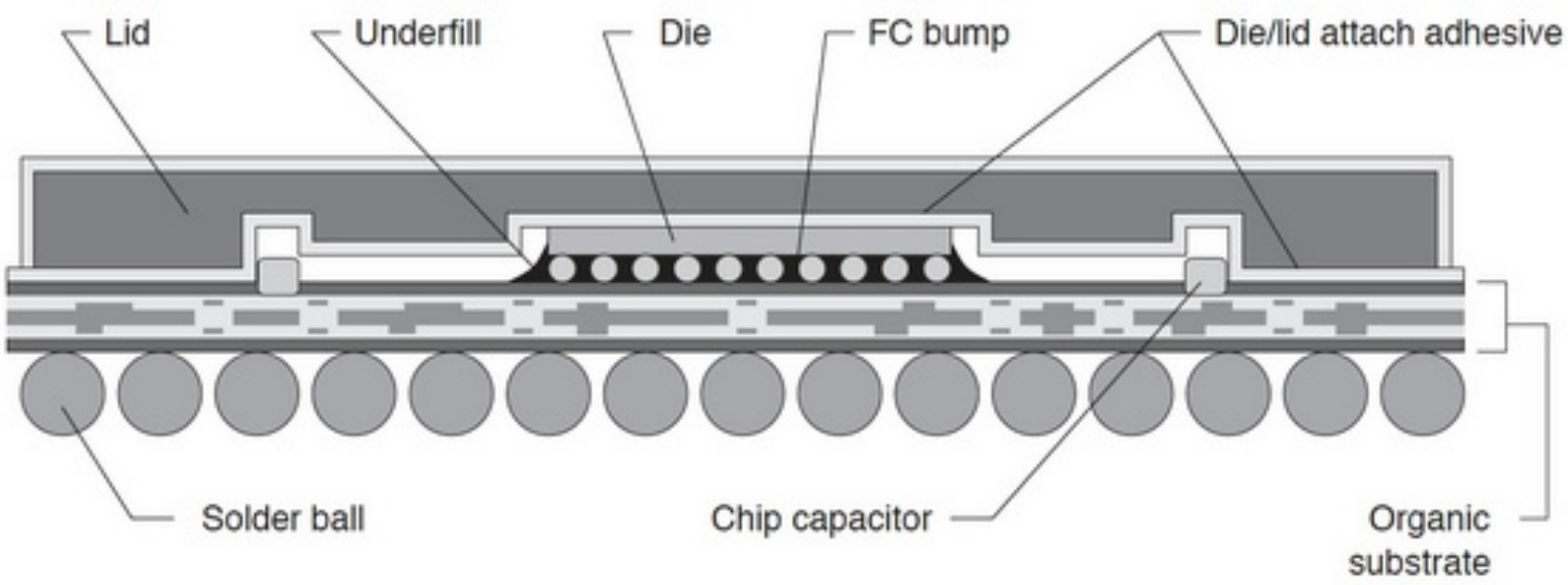


Image from: http://www.shipcircuits.com

Validation

Thermal stressing a PCB

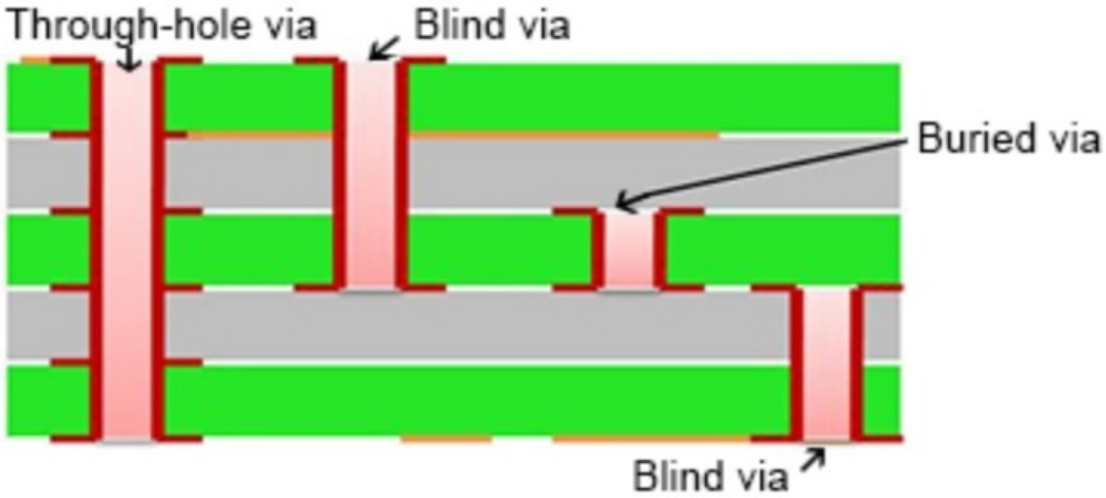


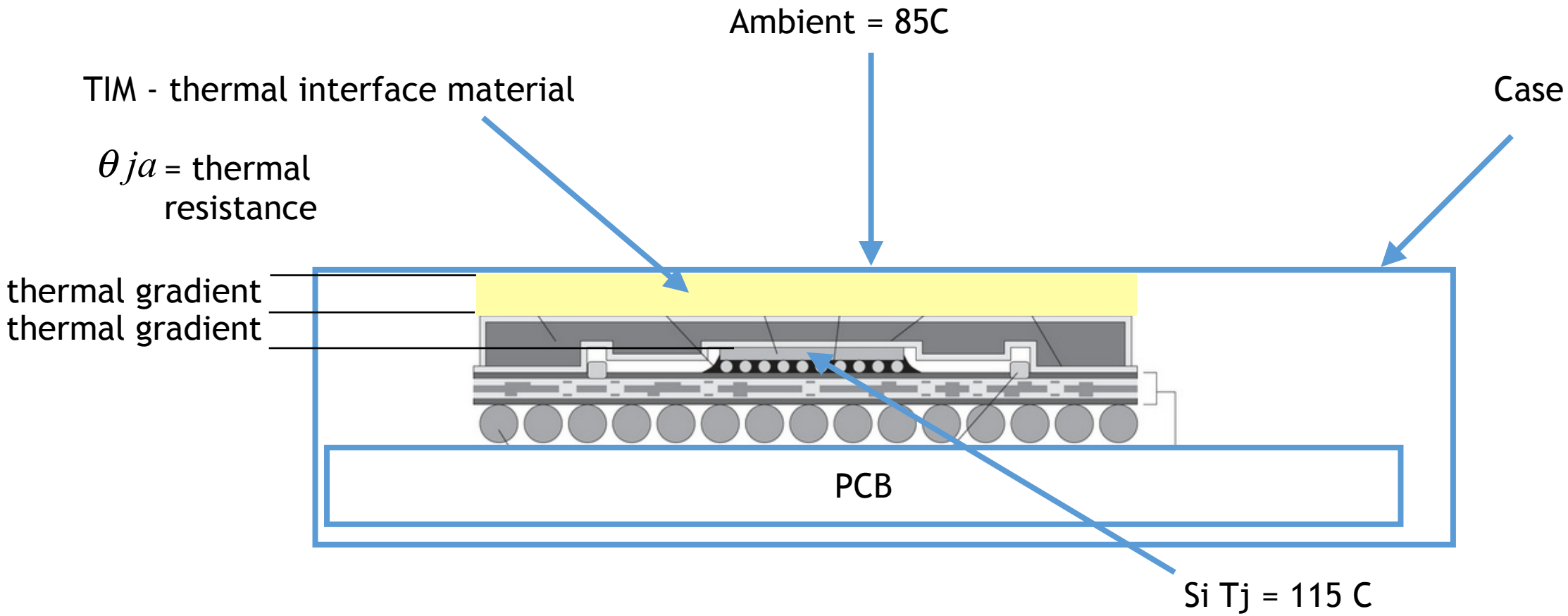
Image from: https://www.quora.com

Temperature Ranges

- Full military (-55˚C to +125˚C)
- Automotive (-40˚C to +125˚C)
- AEC-Q100 Level 2 (-40˚C to +105˚C)
  - A failure mechanism based stress test qualification for packaged integrated circuits.
- Extended Industrial (-40˚C to +125˚C)
- Industrial (-40˚C to +85˚C)
- Commercial (0 C to +85 C)

Sources: https://www.maximintegrated.com/
https://www.altera.com/

Thermal Gradients



Source: https://www.maximintegrated.com/en/app-notes/index.mvp/id/4083

So what about day-to-day activities for an engineer?

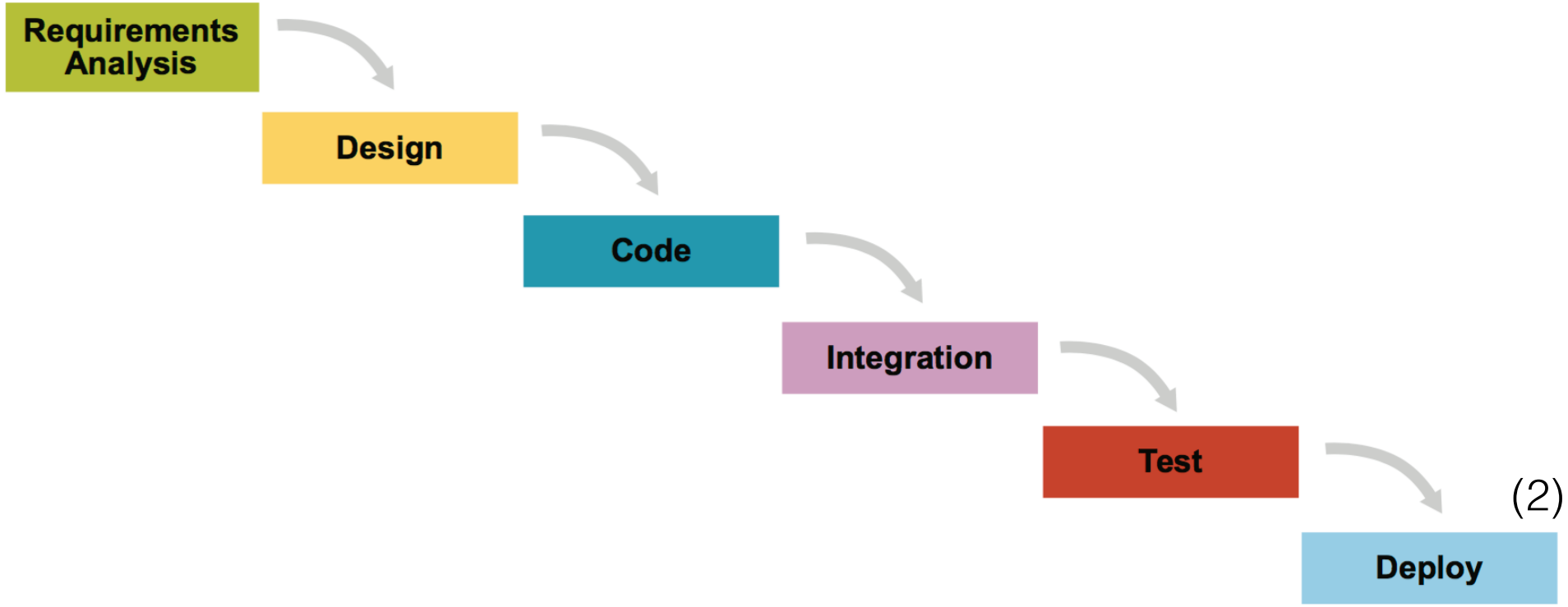
- Agile Development Process

Agile Development Process

- What is Agile?
  - “Agile software development describes a set of principles for software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams.” (1)
- Here is what that means to an engineer:
  - Tasks from the bottom-up schedule are assigned to each engineer
  - Tasks are broken down into smaller sections of work called stories, scrums and sprints. Sprints typically run 2 weeks.

1) https://en.wikipedia.org/wiki/Agile_software_development

Traditional Waterfall Approach



(2)

Figure 1: *Traditional "waterfall" development depends on a perfect understanding of the product requirements at the outset and minimal errors executing each phase.*

(2) http://agilemethodology.org

Agile Approach

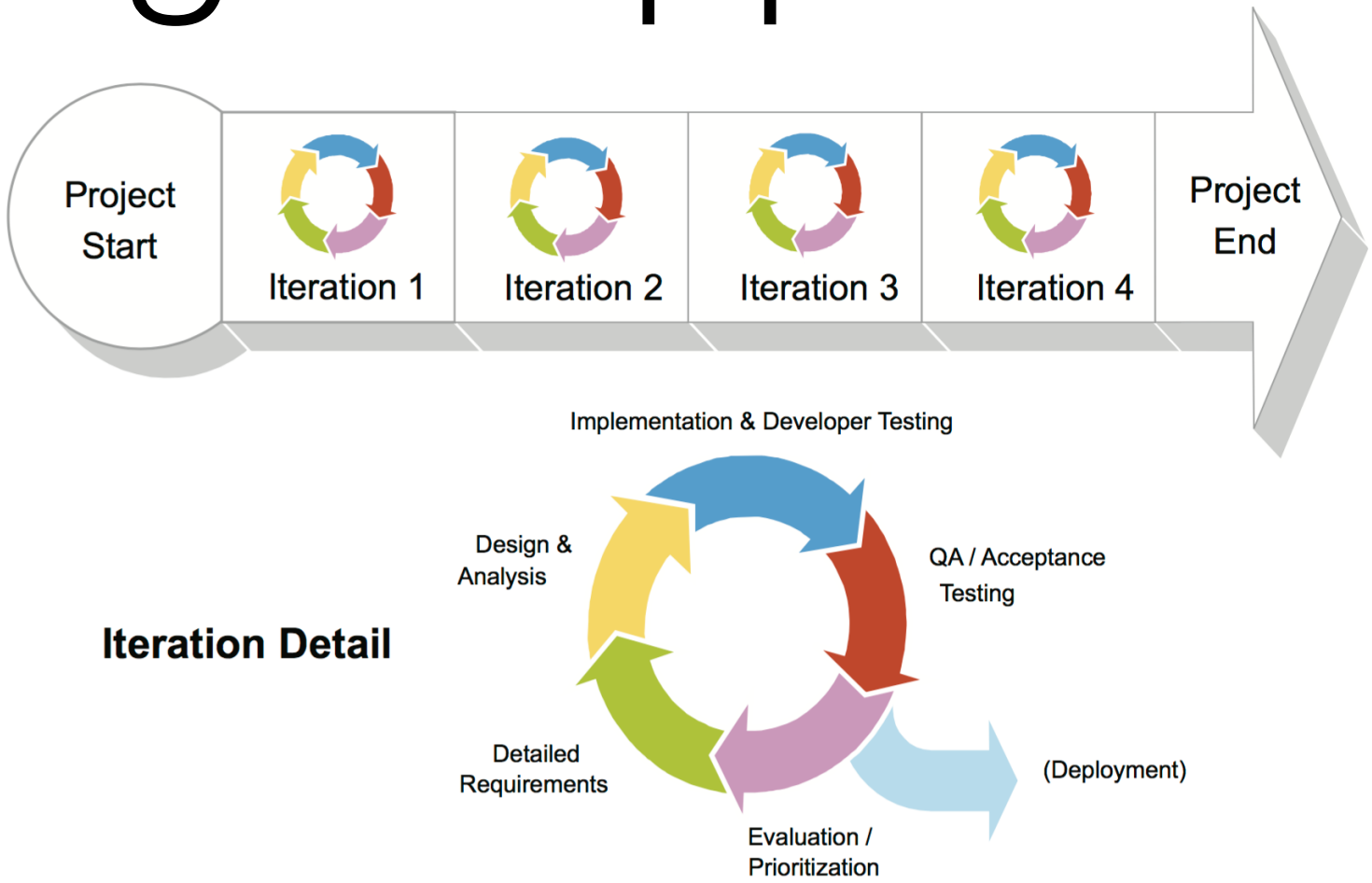


Figure 2: *Scrum blends all development activities into each iteration, adapting to discovered realities at fixed intervals.*

Benefits

- 3D process + Agile development benefits:
  - Projects stay on schedule, fewer schedule slips
    - Builds trust
  - Minimizes development costs
  - Produces higher-quality products
  - Lowers stress on everyone

Documentation

- The importance of documentation in engineering
- You will write a tremendous amount of documentation over the course of your career

Arguments Against

- “Creating all this documentation slows down the creative process”
- “Documentation is a waste of time. Let’s get on and build our next product”
- These mindsets lead to “Tribal Knowledge”
  - Knowledge in employees heads, not written down
  - Over time this reduces an organization’s productivity

Project Management

- Documentation is a central component of Project Management.
- Serves two primary purposes:
  - Capture Requirements
    - “The product shall measure 3.0” X 6.0” 0.2”
    - “The device shall be ready for user interaction in less than 100 ms after power-on.”
  - Define Traceability:
    - What was done?
    - Who did it?
    - When?
    - How?
    - Why?

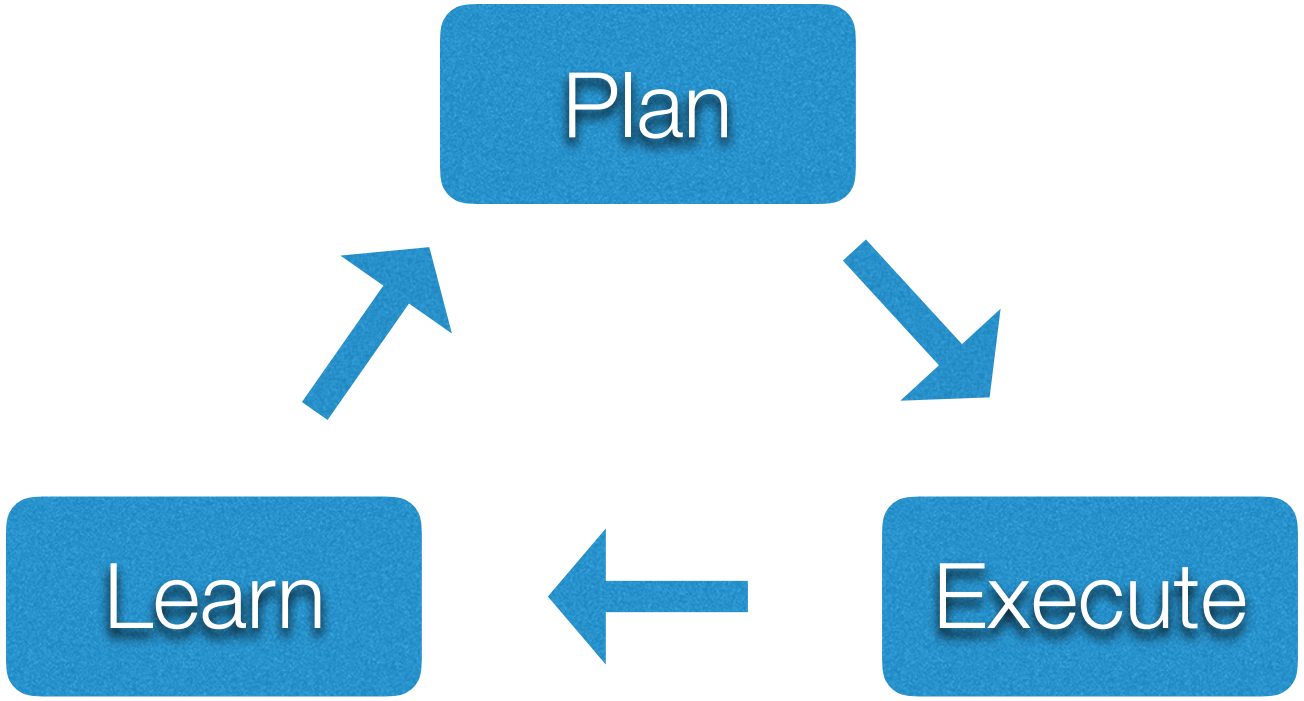
Examples

- Technical feasibility and trade-off studies
  - Consumer: Management team.
- Success metrics
  - Consumer: Management team.
- Requirements
  - Consumer: Implementation and test teams
- Architecture
  - Consumer: Implementation team
- Project schedule, exit criteria, and process flow steps
  - Consumer: Management team (plan resources, manage costs and risks)
- Test plan
  - Consumer: Test and implementation teams

Examples (con't)

- Requirements traceability
  - Consumer: Test, implementation teams, end customer
- Issue/Bug tracking
  - Consumer: Test and implementation teams
- Change management, important to define a "**Change Control Process**"
  - Consumer:Test, implementation and management teams
- Programmers guide
  - Consumer: Software team
- Validation Plan
  - Validation team
- Users guide
  - Consumer: End customer
- Lessons learned
  - Consumer: Everyone on the current project and those on the next project

Project Cycle



Excellent project managers develop a set of document templates for this process and reuse them over and over. Tweaking them as lessons are learned.

Documents

Architecture Definition	Design Planning	Software Development Electronics Development Physical Development	Manufacture	Validation	Production

Feasibility Study

Project Schedule

Success Metrics

Bug Tracking

Programmer's Guide

User's Guide

Lessons's Learned. Feed forward to next project

Requirements

Requirements Traceability

Validation Plan

Bug list Errata

Architecture

Test Plans

Micro Architectures

icons from iconfinder.com and thenounproject.com

Benefits

- Aids everyone on the team, speeds the learning curve for new hires
- Reduces risk. “Tribal knowledge” is “last century.” Too much knowledge in any one person’s head is a risk. Write it down.
- Reduces disagreements and communication mistakes
- Establishes a common set of expectations among stakeholders
- Gives key stakeholders a clear understanding of the project: Scope, Schedule and Resources.
- Enables progress monitoring, “Are we on schedule or not?”

Benefits (con't)

- Helps all team members understand their responsibilities
- Stability and predictability builds trust
- Greatly aids in keeping the end customer up to date
- Certifications such as ISO-9000 require companies to document their process flows
  - Mission Statement, Vision Statement, Quality Policy
- Excellent communication and writing skills can "sell" at a job interview

Building a Bill of Materials

Building a Bill of Materials

- A bill of materials (BOM) explains what to buy and how many to buy. It also describes the cost of materials for a product.
- Management and engineering use this information to assess a product's price competitiveness and to set the price margin for the product. Example:
  - Suggested retail price: BOM cost * 5
  - Reseller (NewEgg, Amazon) price: BOM cost * 2.5
  - Distributor (Digikey, Arrow Electronics) price: BOM cost * 1.75
- $Revenue = (units sold * reseller price) + (units sold * dist. price)
- $Profit = $Revenue - (BOM cost * units sold) - (CapEx + OpEx)

CapEx = Capital expenditures: PC's, lab equipment, software. OpEx = Operational expenditures: heat, light, rent

Building a Bill of Materials

- See example BOM .xlsx file

Product Teardown Analysis

- The process of taking a product apart to learn the list of components that make up the product
- Cost estimates of the individual components can then be made
- Functional diagrams (block diagrams, schematics) can then be drawn to gain insight into how the product operates

Product Teardown Analysis

- Resources:
  - http://www.bcg.com/expertise/capabilities/procurement/product-teardown.aspx
  - https://seniordesign.engr.uidaho.edu/processdocs/teardown.pdf
  - Google “Product teardown”

End