Protótipo; Testes; V&V

PSI2594 Projeto de Formatura II

2o. Semestre 2016

Material

- Ford&Coulston

Motivação

- Desenvolvimento é acompanhado de “bugs.”
- Descobrir os “bugs” mais cedo representa menos custos
  - O impacto no sistema será maior se o bug for descoberto só nas fases avançadas do desenvolvimento
    - Uma correção de bug requer que todos os módulos relacionados sejam retestados.
    - Uma correção de bug requer reprojeto dps módulos relacionados.
  - Por exemplo
    - Erro na Placa de Circuito Impresso
    - Erro de leiaute VLSI
    - Um erro sutil de código
- Teste não remove bugs, apenas torna menos provável a sua existência.

Categorias de Protótipos

- Proof-of-Principle
- Form Study Prototype
- User Experience Prototype
- Visual Prototype
- Functional Prototype

Protótipo x Projeto (design) de produção

- Materiais
- Processos
- Baixa Fidelidade

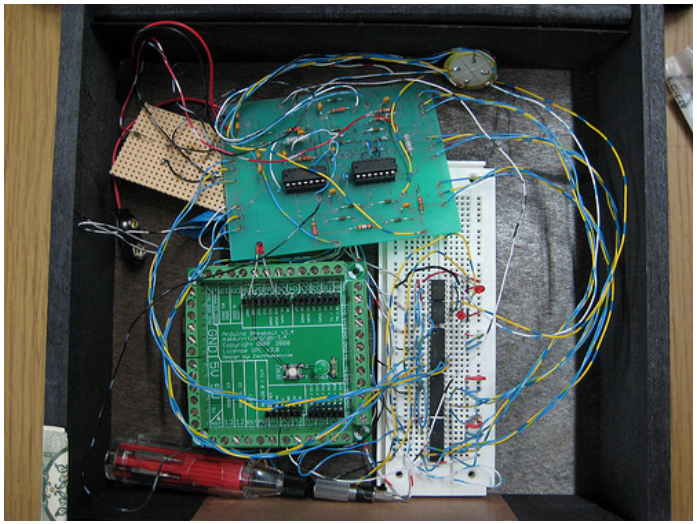
Protótipo em Engenharias

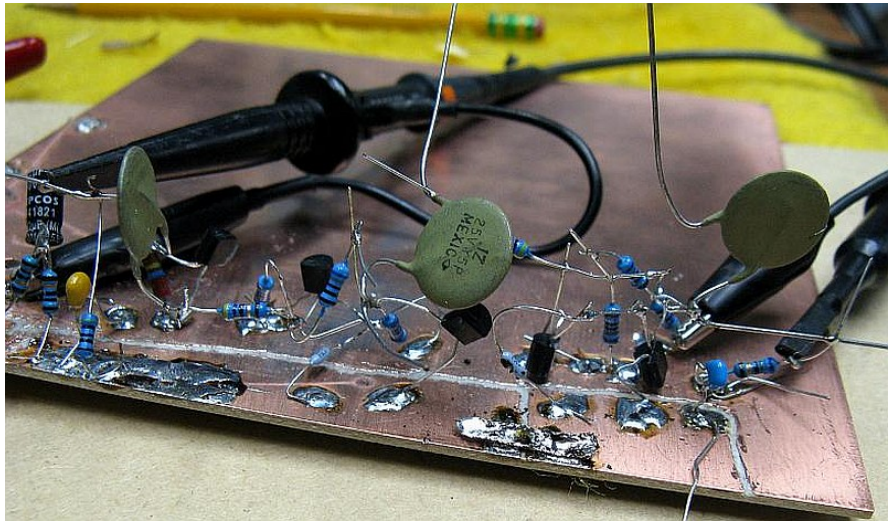
- Um protótipo grosseiro é construído como prova de conceito.
- Uma aplicação de patente frequentemente requer uma demonstração de funcionalidade antes de ser preenchido.
- Universidades tem Centros de Prova de Conceito

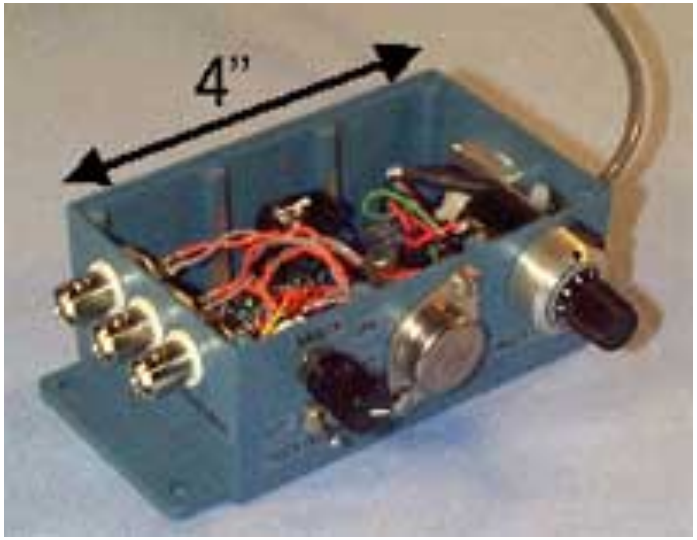
O que será o resultado do Projeto de Formatura?

- Protótipo prova de conceito?
  - Versão preliminar para provar um conceito, sem preocupação com materiais, processos, aspectos mecânicos e estéticos
- Protótipo funcional?
  - Versão operacional (working) mas levando em conta alguns aspectos mecânicos e estéticos
- Produto?
  - Envolve os processos de produção, entre outros

11/09/13





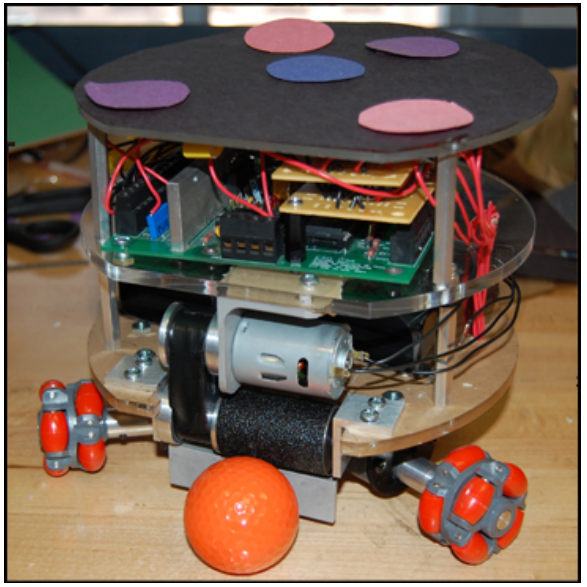


Como serão verificados os requisitos?

- Funcionais
- Não Funcionais
  - Faixa de temperatura de operação
  - Vibração
  - Etc.

- O protótipo prova de conceito ou o funcional é suficiente?
- Como fica a aderência a normas?
  - Por exemplo: Eletromagnética EMI&EMC

O que se espera....



Lembrem-se

- Papel do engenheiro
  - Fazer o design de produtos
- Dois tipos de projeto
  - Pesquisa – gera conhecimentos
    - Pesquisa básica
      - Geral
      - Dirigida
    - Aplicada
  - Desenvolvimento
- Inovação
  - Não vale se não for ao mercado

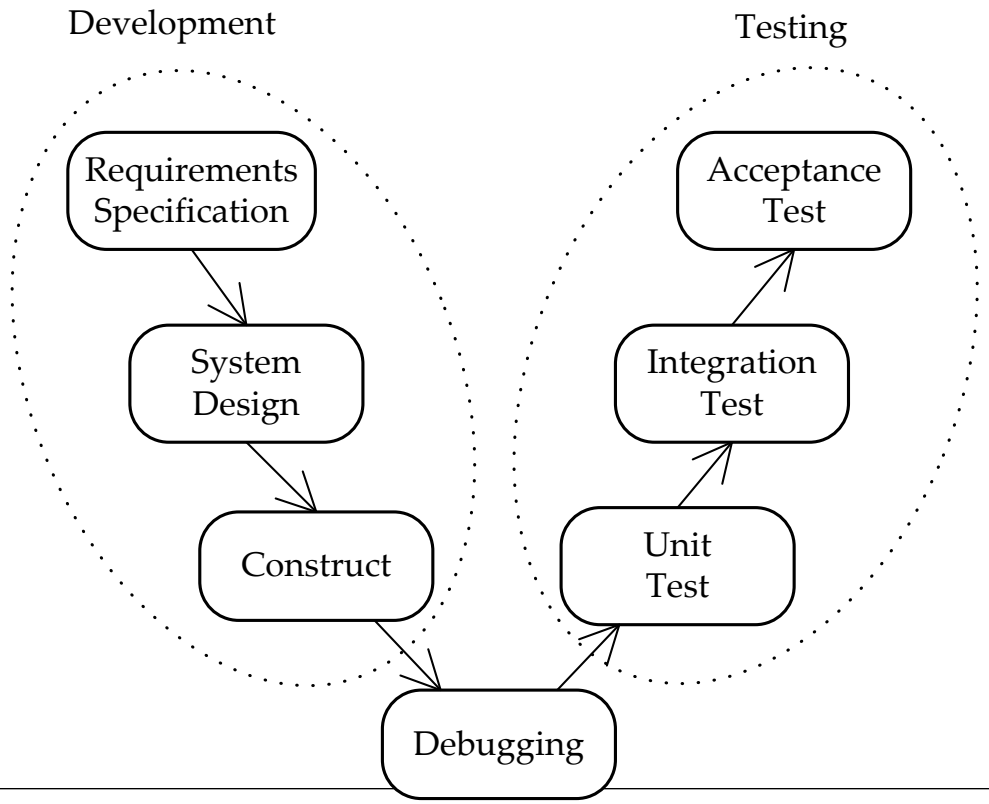
Lembrem-se...

- TCC Projeto de Engenharia
  - Algo entre protótipo e produto
  - Não é uma "feira de invenção"
  - Se for protótipo, tem de mostrar atributos de manufacturabilidade e total aderência aos requisitos funcionais e não funcionais
  - Se for produto, melhor ainda
    - Envolve circuitos impressos? Terceirize
    - Envolve chassis mecânico? Terceirize
    - Etc.

Plano de Teste

1 Princípios de Teste

- Teste acontece ao longo do processo de projeto
  - Escreva testes junto com o projeto dos módulos,
  - Realize testes enquanto estiver implementando os módulos
- O diagrama Test-V ilustra este processo



Teste

- Documentos de qualidade de projeto (design)
  - Requer você raciocinar sobre o comportamento do sistema
  - Oferece a você metas claras de projeto
  - Olhar sobre o sistema sob o ponto de vista do testador
- Como devemos realizar o teste?
  - Compromisso entre
    - Aplicar todas as entradas possíveis, ou
    - Só aplicar entradas selecionadas que possam produzir erros
  - Teste deve ser escolhido de forma a aumentar as chances de se encontrar erros.

Porque Casos de Teste?

- Casos de Teste definem exatamente o que o módulo deve fazer.
- Casos de Teste forçam o projetista a pensar nos casos extremos.
- Casos de Teste forçam o projetista a repensar o projeto do módulo antes de construí-lo.

Tipos de teste

- Black box
  - Nenhum conhecimento da organização interna
  - Acesso apenas às entradas e saídas
  - Altera-se as entradas e se observa as saídas
- White box
  - Conhecimento da organização interna
  - Pode ter expectativa de modelo de falha
  - Criar instancias de teste que revelem erros lógicos ou físicos

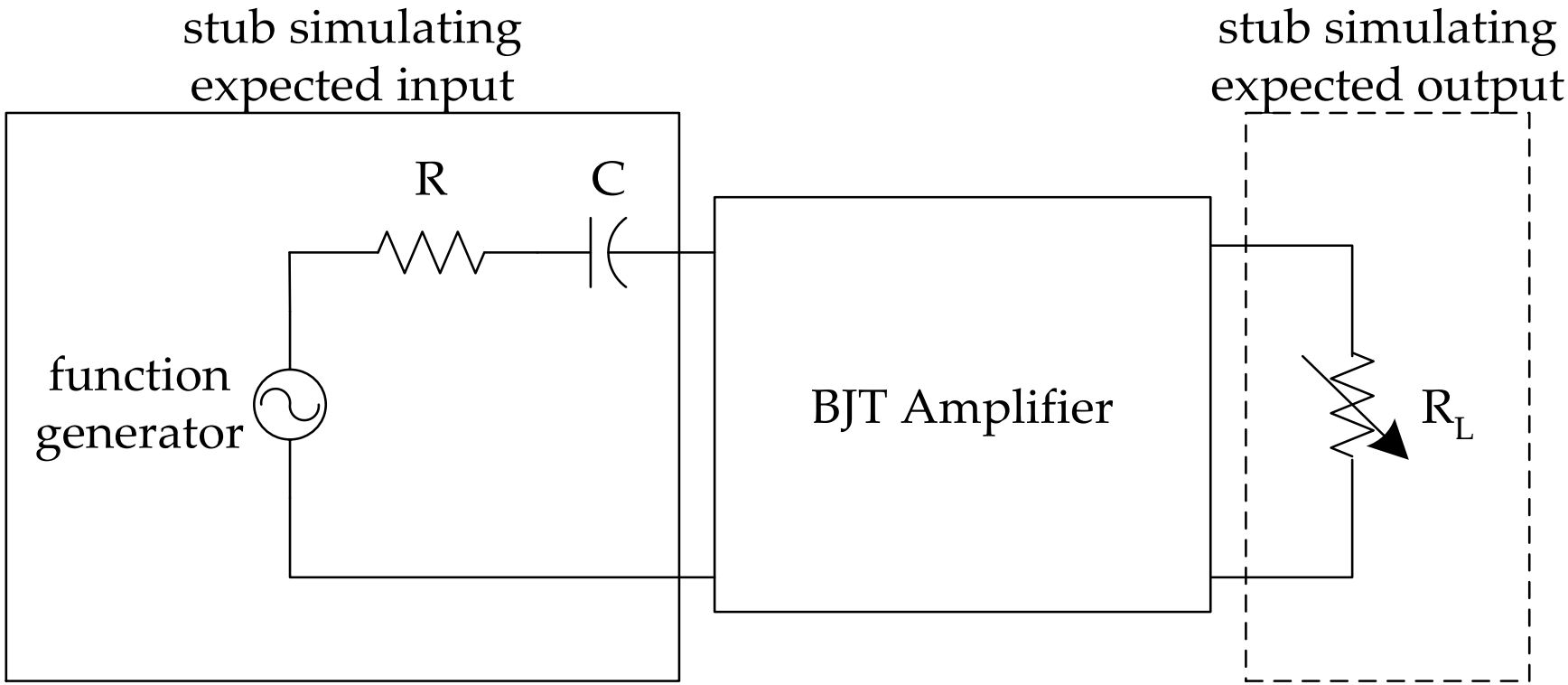
Testabilidade

- Componente Testável – uma falha em um componente pode ser
  - Rapidamente detectada
  - Rapidamente localizada
- Controlabilidade
  - Quando qualquer nó de um sistema pode ser colocado em um estado desejado
  - Black box não tem controlabilidade
- Observabilidade
  - Quando qualquer nó do sistema pode ser medido.
  - Black box tem baixa observabilidade

Stub

- Stub
  - Um espaço reservado para uma futura funcionalidade
  - Um dispositivo que faz mimica de um subsistema
    - Simula entradas ou monitora saídas
    - Para uma Unidade sob Teste - unit under test (UUT)
    - Assegura bom comportamento
      - Antes de provocar estragos no resto do sistema
- Por exemplo
  - Um gerador de função para uma entrada de áudio
  - Um printf() em vez de uma escrita em arquivo
  - Um DIP switch em vez de uma conexão barramento

Exemplo de Stub



Propriedades dos casos de teste

- *Acurado* - The test should check what it is supposed to and exercise an area of intent.
- *Economico* - The test should be performed in a minimal number of steps.
- *Limitado em complexidade* - Tests should consist of a moderate number (10-15) of steps.
- *Repetitível* - The test should be able to be performed and repeated by another person.
- *Conveniente* - The complexity of the test should be such that it is able to be performed by other individuals who are assigned the testing task.
- *Rastreável* - The test should verify a specific requirement.
- *Self cleaning* - The system should return to the pre-test state after the test is complete.

2 Construindo Testes

Debugging

- Bohrbugs
  - Bohrbugs são bugs confiáveis
  - O erro está sempre no mesmo local
  - Analogia – um elétron tendo uma posição definida
  - Solução = monte uma boa armadilha (set a good trap)
- Heisenbugs
  - Mudanças inócuas da entrada produzem comportamento errático (buggy behavior)
  - Pode não ser reproduzível
  - Parecem estar se movendo ao longo do sistema
  - Analogia – um elétron é uma função densidade "esquiva"
  - Solução = pense diferente, de maneira não convencional (think outside the box)

Processo de Debugging

- Observe o problema sob diferentes condições operacionais
- Formule hipóteses sobre os problemas potenciais
- Conduza experimentos para confirmar ou eliminar as sobre as fontes hipotéticas do problema
- Repita até que o problema seja eliminado

Problemas Comuns

- Cheque os problemas mais fáceis primeiro
- Comece no nível de abstração mais baixo
- Exemplos
  - O sistema está alimentado corretamente?
  - O equipamento de teste está ajustado corretamente?
  - As linhas de barramento (comunicação) estão sendo manipuladas corretamente?
  - Você inicializou o sistema?
  - Você está imprimindo a variável/tipo correto?

Teste de Unidade

- Um teste de unidade é um teste de uma funcionalidade de um módulo de sistema isolado
- Deve ser rastreável até o projeto detalhado (detailed design).
- Consiste de um conjunto de casos de teste
- Cada caso de teste estabelece que um subsistema realiza uma simples unidade de funcionalidade para alguma especificação.
- Casos de teste devem ser escritos com a intenção expressa de descobrir defeitos não descobertos.

Teste de Unidade – cont.

- *Write unit test* durante a implementação; porque?
  - *White box tests*
  - Pensar a respeito das situações e condições de teste podem auxiliar o projetista a descobrir erros antecipadamente.
  - Testes de unidade precisam ser escritos com um entendimento da organização interna do componente, e um sistema é melhor entendido durante seu desenvolvimento.

Teste de unidade - exemplo

```
if (16 < Celsius Temperature < 32)
    Fahrenheit Temperature = ROM[input-16];
else
    Fahrenheit Temperature = (9* Celsius Temperature)/5 + 32;
```

- Que entradas são usadas para checar a ROM?
- Que entradas checam as condicionais?
- Caminho de Processamento – sequencia de A a B
  - Cobertura de Teste
  - Cobertura completa de caminho

Código exemplo

- Determine as entradas para checar todos os caminhos
  - ACTION1
  - ACTION2
  - ACTION3

```
if ((x >= 30) && (x <= 39)) {
    ACTION1
} else if ((x >= 80) && (x <= 96)) {
    ACTION2
} else if ((x >= 40) && (x <= 56)) {
    ACTION 3
}
```

Métodos de Teste

- Matriz de Teste
- Testes automatizados por scripts
- Teste passo-a-passo

Nota: estes métodos podem ser aplicados a qualquer nível de teste: unidade, integração, aceitação, etc.

Matriz de teste

Teste passo-a-passo

Teste de Integração

- Escreva o teste de integração durante o projeto (design) de nível 1
  - Ajuda a assegurar que os requisitos sejam atingidos.
  - Ajuda a firmar o projeto (design)
  - Requer que o projetista pense a respeito do comportamento dos subsistemas.
  - Requer que o projetista pense sobre os comportamentos extremos dos subsistemas.

Escreva o Teste de Integração

- Quais são os caminhos diferentes de execução através do sistema?
- Todos os módulos forma exercitados pelo menos uma vez durante o teste de integração?
- Todos os sinais de interface foram testados?
- Todos os modos de interface foram exercitados?
- O sistema processa informação na taxa requerida e atingiu os requisitos de temporização?

Testes de Aceitação

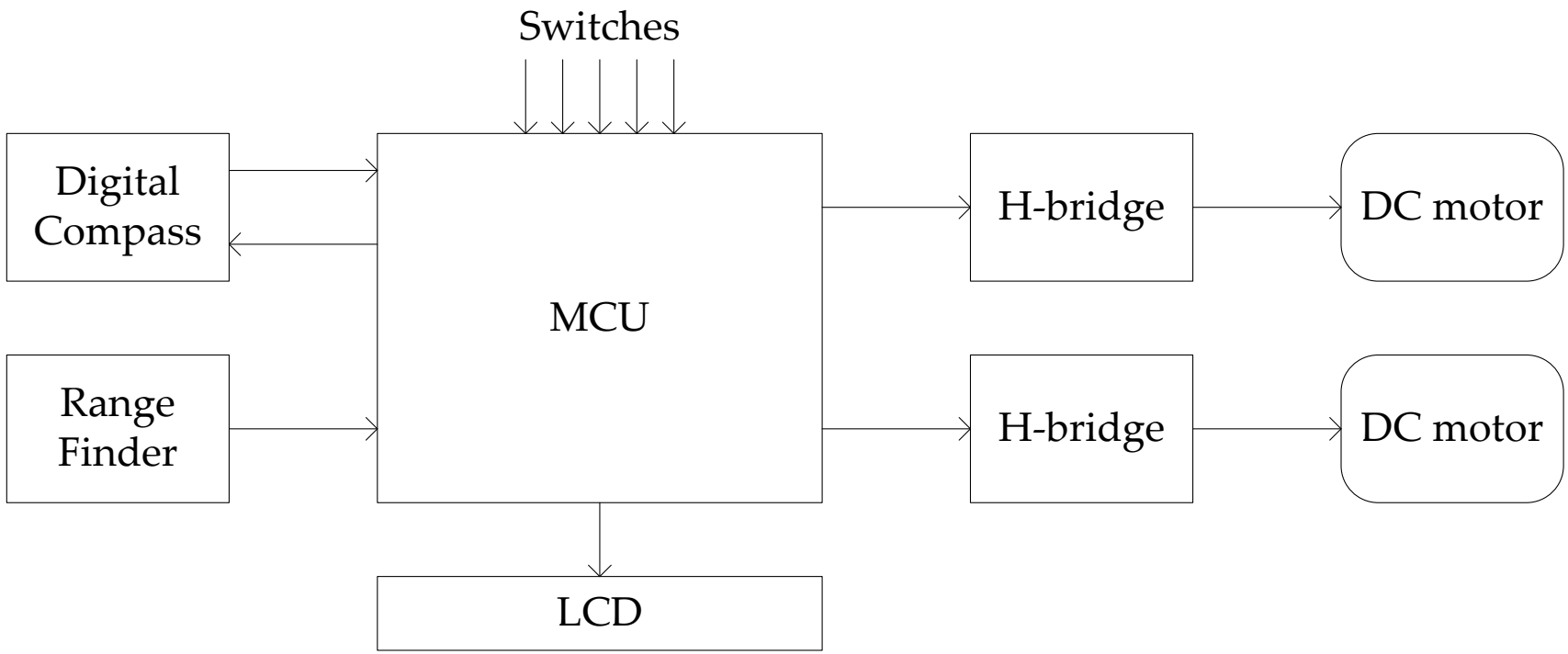
- Pode ser um documento formal legal
- Escrito juntamente com os requisitos
- Rastreável aos requisitos de engenharia
- Identifica
  - Escopo– Quanto do sistema é testado?
  - Nível – a profundidade que o teste deve ser realizado

3 Aplicação: Robot Autônomo

- Autonomous navigating robot
- Engineering requirements
  - *The robot's center must stay within 12 to 18 centimeters of the wall over 90% of the course, while traveling parallel to a wall over a 3 meter course.*
  - *The robot's heading should never deviate no more than 10 degrees from the wall's axis, while traveling parallel to a straight wall over a 3 meter course.*

Teste de aceitação do Robot

Exemplo: Arquitetura do Robot



Alguns possíveis testes de integração

- MCU + motors + bridge + switches
- Chassis + digital compass + MCU + motors + bridge +LCD
- Chassis + range finder + MCU + motors + bridge

Teste de integração passo-a-passo

Possibilidades de teste de unidade

- MCU (hardware)
- LCD
- Switches
- Compass
- Range finder
- H-bridge
- Motors
- Chassis
- MCU (software)

Unit Test: The Digital Compass

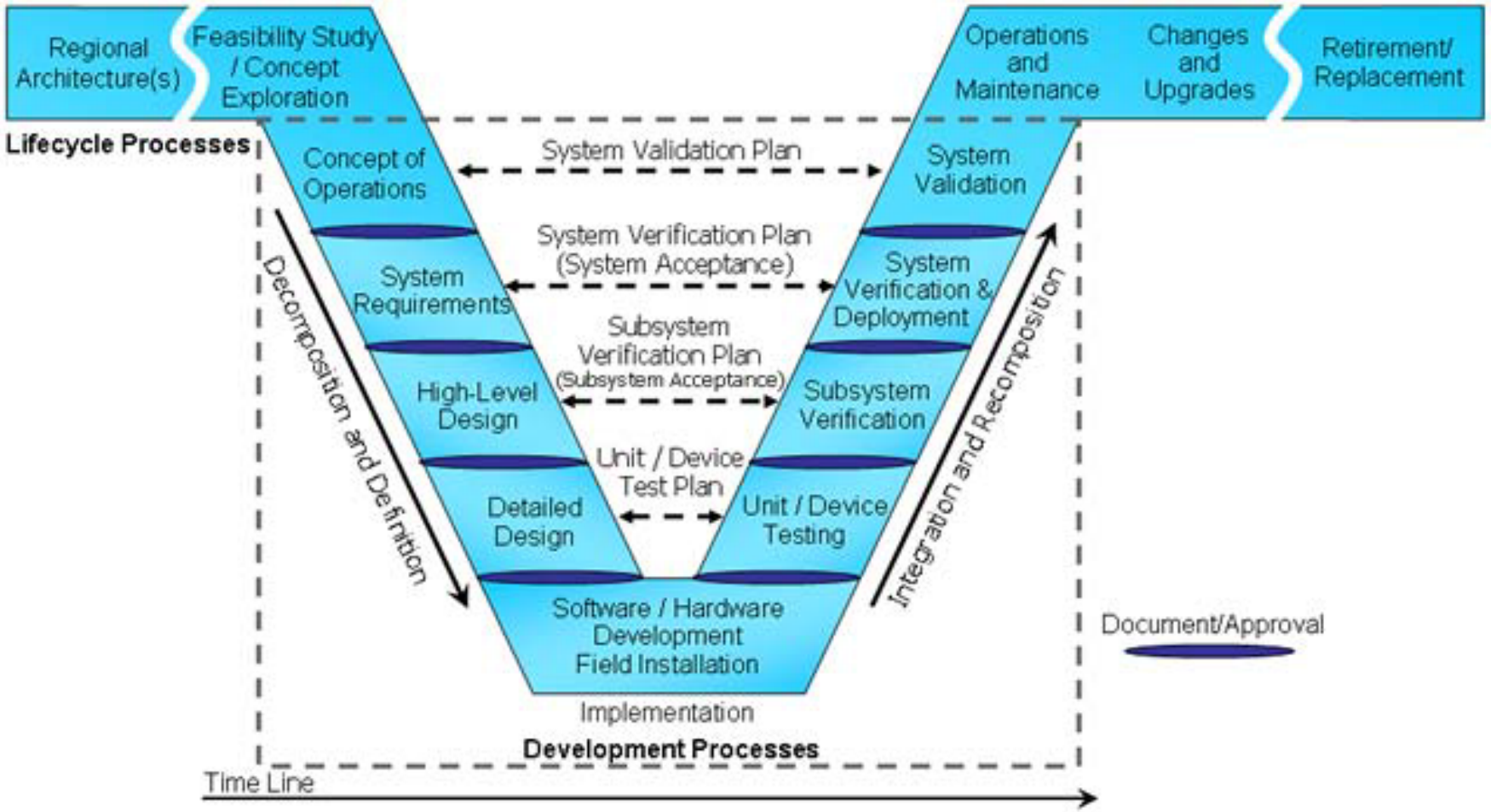
Unit test: compass (matrix)

4 Diretrizes

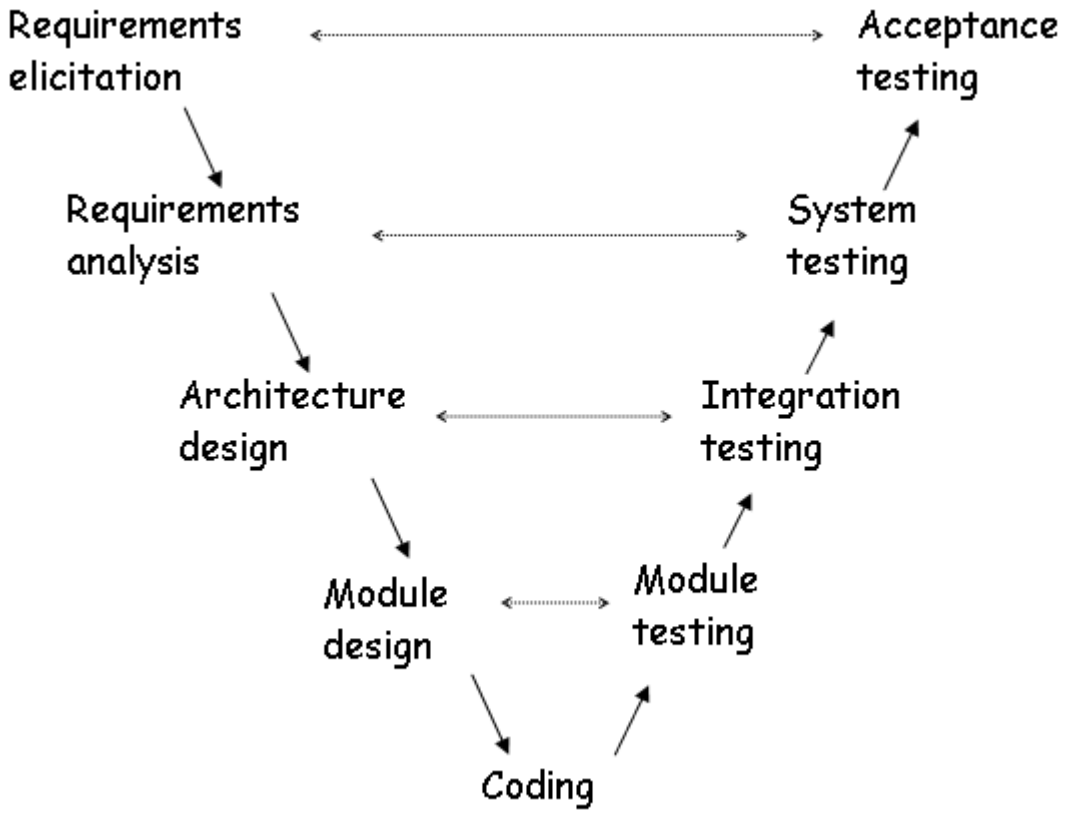
Razões para desenvolver e conduzir testes

- Testes reduzem o numero de bugs.
- Testes constituem-se em boas documentações.
- Testes melhoram o projeto (design).
- Testes permitem a refatoração.
- Testing força você a desacelerar e pensar.
- Teste torna o desenvolvimento mais rápido.
- Testes aumentam a segurança (confidence) no projeto.

Outros: V&V


Regional Architecture(s)
Feasibility Study / Concept Exploration
Operations and Maintenance
Changes and Upgrades
Retirement/ Replacement
Lifecycle Processes
Concept of Operations
System Validation Plan
System Validation
System Verification Plan (System Acceptance)
System Requirements
System Verification & Deployment
Subsystem Verification Plan (Subsystem Acceptance)
High-Level Design
Subsystem Verification
Unit / Device Test Plan
Detailed Design
Unit / Device Testing
Decomposition and Definition
Integration and Recomposition
Software / Hardware Development Field Installation
Document/Approval
Implementation
Development Processes
Time Line

V&V em software



Abordagem do Modelo V

- Abordagem top-down no braço esquerdo do V
- Abordagem bottom-up do braço direito do V

- Explique!