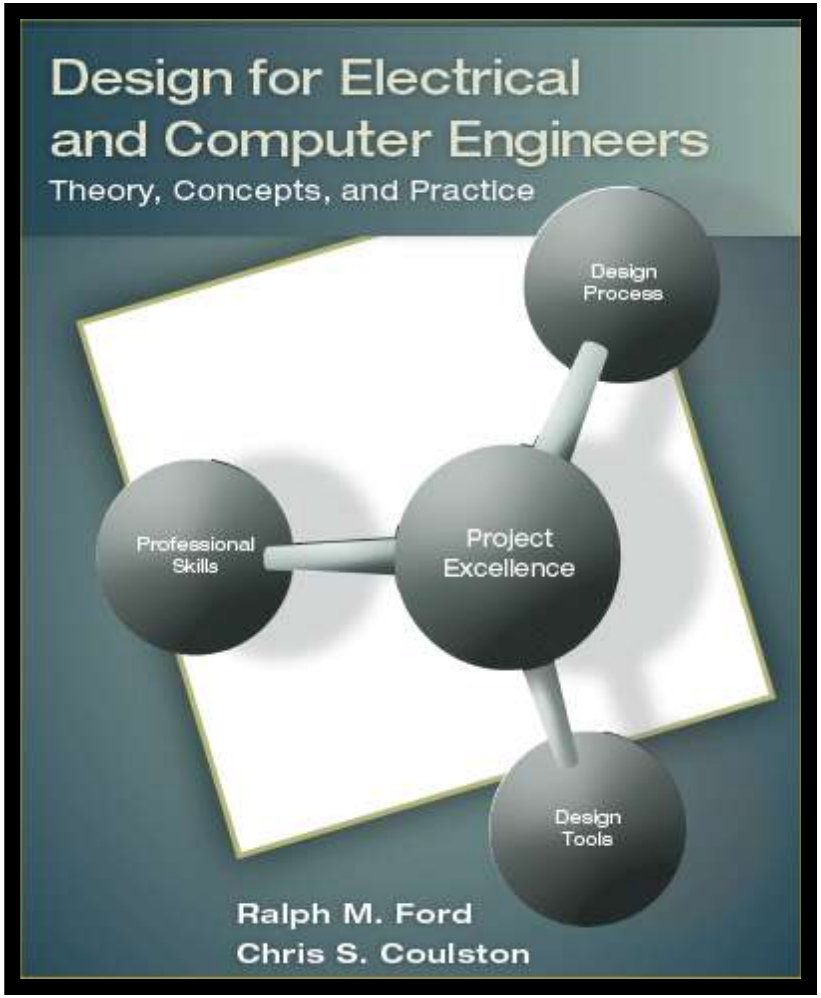
Aula04 Protótipo; Testes; V&V

PSI2594 Projeto de Formatura II

2o. Semestre 2011

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 leioute VLSI
    - Um erro sutil de código
- Teste nãoremove 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

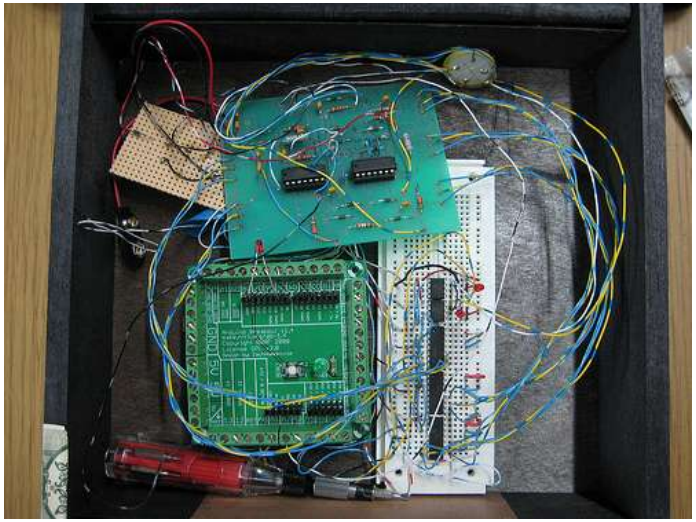
- Materiais
- Processos
- Baixa Fidelidade

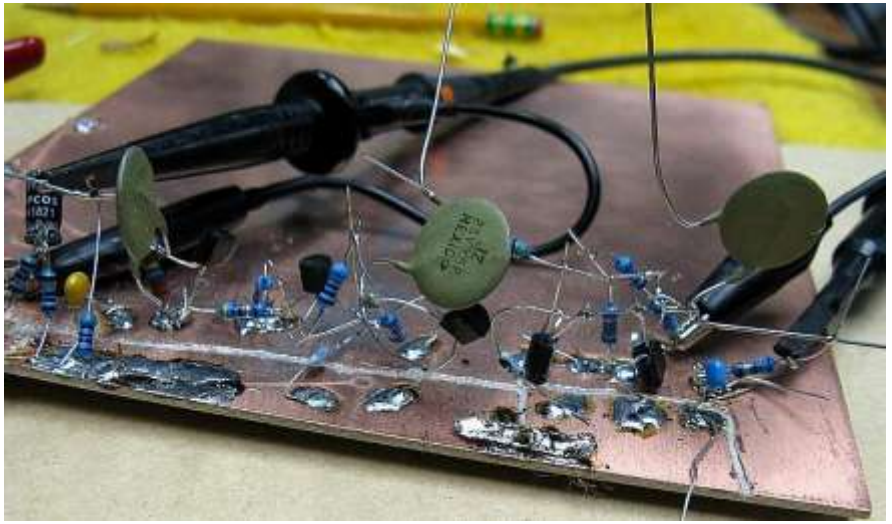
Protótipo em Engenharias

- Um protótipo grosseiro é contruido como prova de conceito.
- Uma aplicação de patente frquentemente 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 mecanicos e estéticos
- Protótipo funcional?
  - Versão operacional (working) mas levando em conta alguns aspectos mecanicos e esteticos
- Produto?
  - Envolve os processos de produção, entre outros





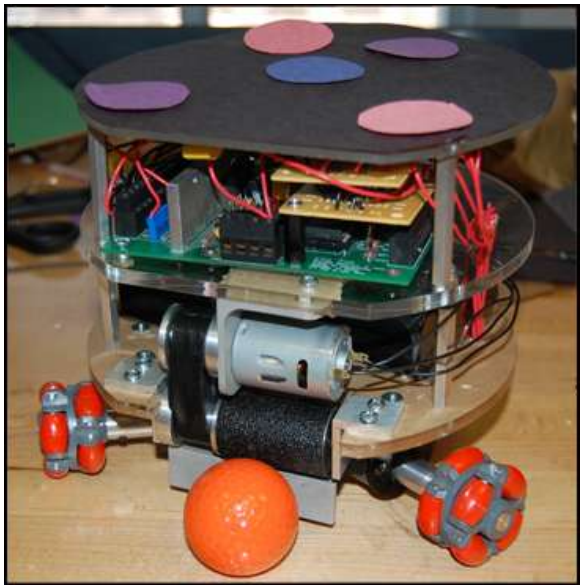
02/09/2011

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 aderencia a normas?
  - Por exemplo: Electromagnetica 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

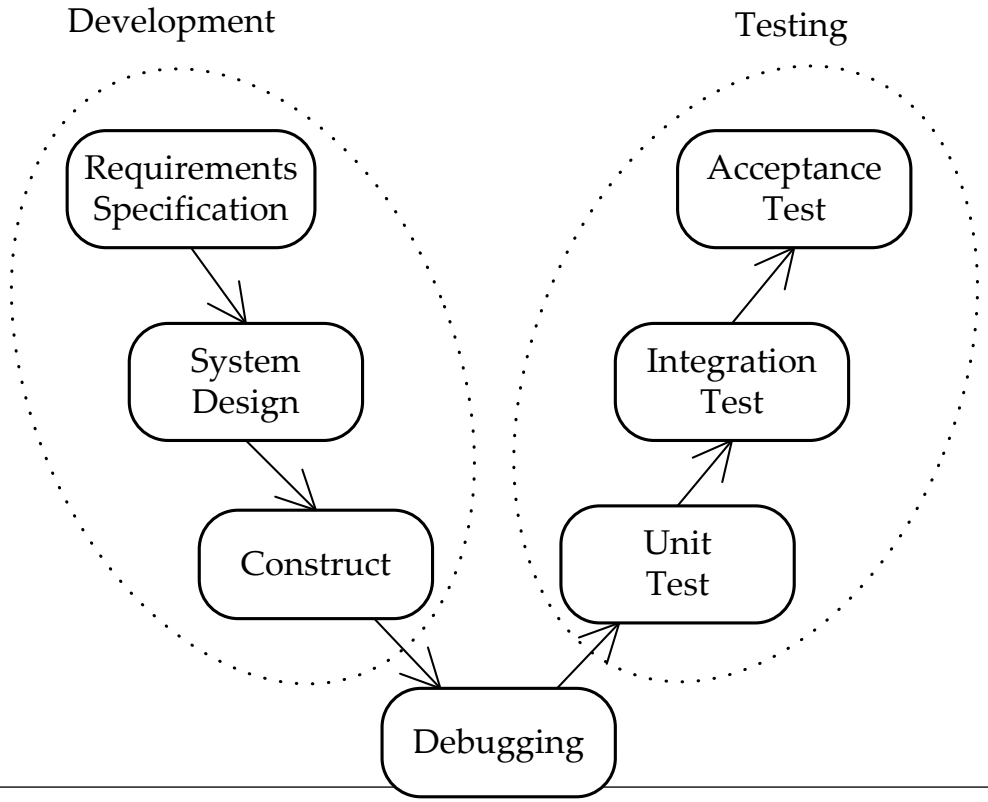
Lembrem-se…

- 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 aderencia aos requisitos funcionais e não funcionais
  - Se for produto, melhor ainda
    - Envolve circuitos impressos? Terceirize
    - Envolve chassis mecanico? 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-Vee ilustra este processo



Teste

- Documentos de qualidade de projeto (design)
  - Requer voce raciocinar sobre o comportamento do sistema
  - Oferece a voce 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 possiveis, 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

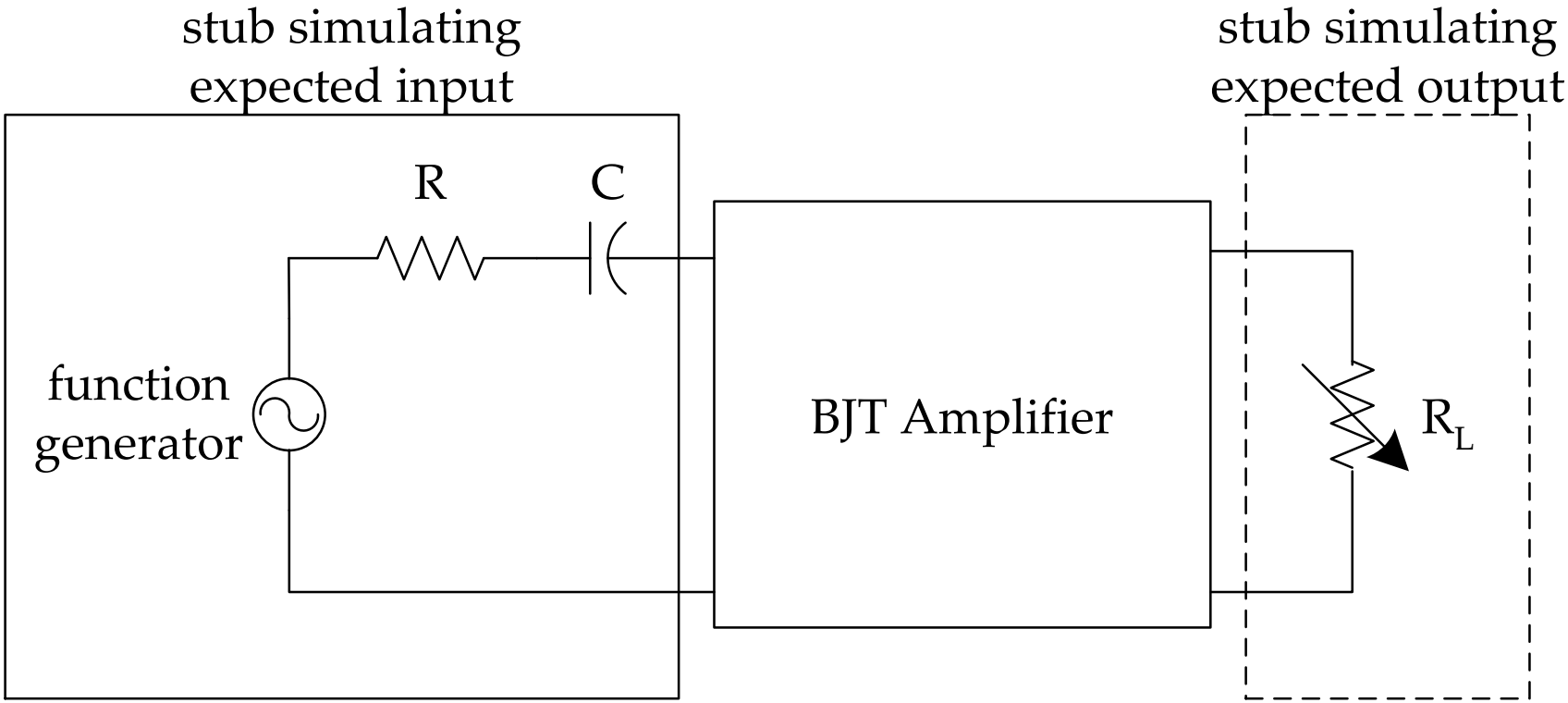
Testabilidade

- Componente Testavel – uma falha em um componente pode ser
  - Rapidamente detectada
  - Rapidamente localizada
- Controlabilidade
  - Qdo. Qq. Nó de um sistema pode ser colocada em um estado desejado
  - Black box não tem controlabilidade
- Observabilidade
  - Qdo. Qq. 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.
- *Limitedo 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 de 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 hipoteses sobre os problemas potenciais
- Conduza experimentos para confirmar ou eliminar as sobre as fontes hipotéticas do problema
- Repitas até que o problemas 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?
  - Voce inicializou o sistema?
  - Voce 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 during implementation; why?
  - White box tests
  - Thinking about test situations and conditions may lead the designer to uncover errors before they are ever designed into the system.
  - Unit tests need to be written with an understanding of the internal organization of the component, and a system is best understood during its development.

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 inputs to check all paths
  - 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 qq nível de teste: unidade, integração, aceitação, etc.

Matriz de teste

Test Writer: Sue L. Engineer								
Test Case Name:	ADC function test			**Test ID #:**	ADC-FT-01			
Description:	Verify conversion range and clock frequency. Output goes to 0 in presence of null clock.			**Type:**	☐ white box ☑ black box			
Tester Information								
Name of Tester:				**Date:**				
Hardware Ver:	1.0			**Time:**				
Setup:	Isolate the ADC from the system by removing configuration jumpers.							
Test	**V_T**	**Clock**	**Expected output**		**Pass**	**Fail**	**N/A**	**Comments**
			Decimal	**Hexadecimal**				
1	0.0V	10kHz	0	0x000				
5	2.0V	0Hz	0	0x000				
Overall test result:								

Teste passo-a-passo

Test Writer: Sue L. Engineer					
Test Case Name:		Finite State Machine Path Test #1		**Test ID #:**	FSM-Path-01
Description:		Simulate insertion of money with a mix of nickels and dimes. Verifies FSM, outputs candy in response to a total deposit of $0.30.		**Type:**	☑ white box ☐ black box
Tester Information					
	Name of Tester:			**Date:**	
Hardware Ver:		1.0		**Time:**	
Setup:		Make sure that the system was reset sometime prior and is in state $0.00.			

Ste p	Action	Expected Result	Pas s	Fail	N/A	Comments
1	Strobe Nickel	State should go to $0.05				
2	Strobe Dime	State should go to $0.15				
3	Wait	State should remain $0.15				
4	Strobe Nickel	State should go to $0.20				
5	Strobe Dime	State should go to $0.25				
6	Nothing	State should go to $0.00				
Overall test result:						

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 Autonomo

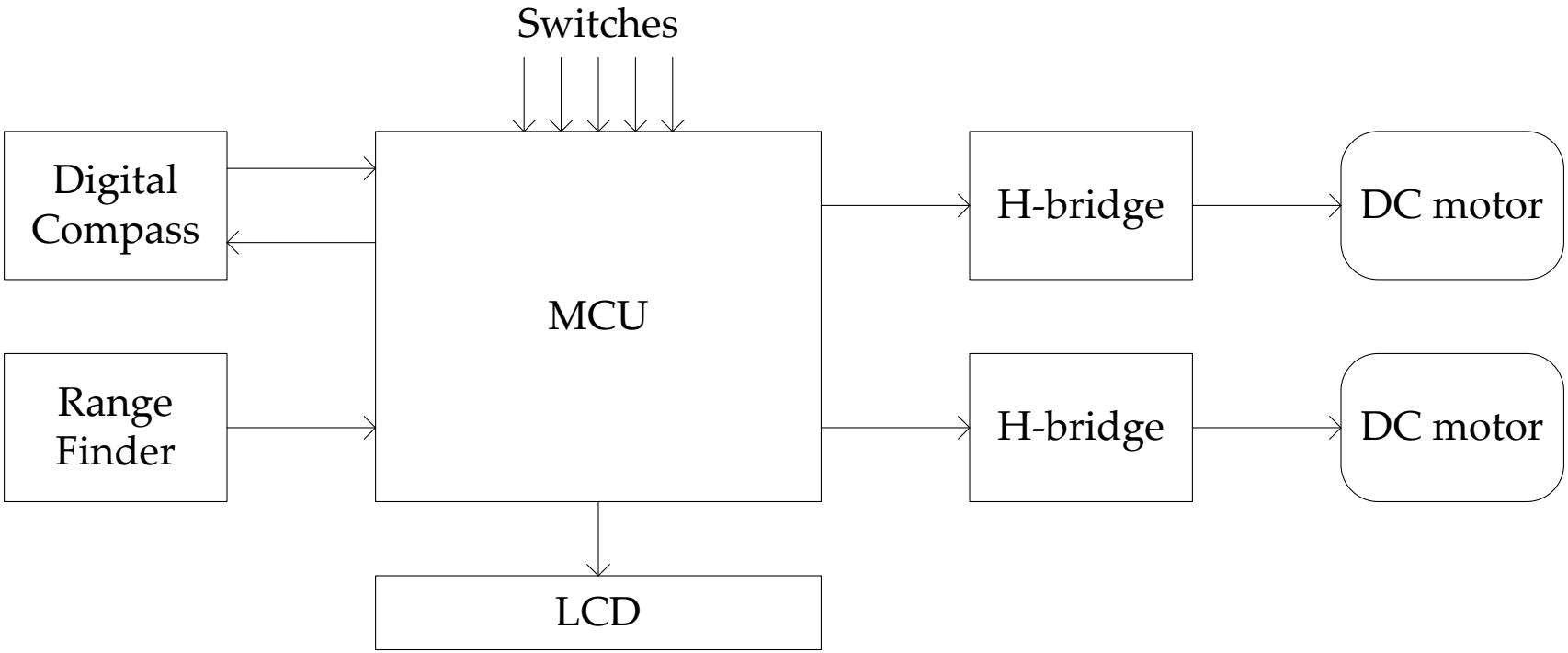
- Autonomous navigating robot
- Engineering requirements
  - *The robot's center must stay within 12 to 18 centimeters of the wall over 90% of the course, while traveling parallel to a wall over a 3 meter course.*
  - *The robot's heading should never deviate no more than 10 degrees from the wall's axis,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

Teste de aceitação do Robot

Test Writer: Sue L. Engineer					
	Test Case Name:	Robot acceptance test #1		**Test ID #:**	Robot-AT-01
	Description:	Checks the engineering requirement: *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.*		**Type:**	☐ white box ☑ black box
Tester Information					
	Name of Tester:			**Date:**	
	Hardware Ver:	Robot 1.0		**Time:**	
	Setup:	Completed robot should be fully charged and placed on 3 meter test track.			

Step	Action	Expected Result	Pass	Fail	N/A	Comments
1	Write a program to monitor the robots position from the wall.	Program should be statically tested to verify accuracy. Should sample wall at a sufficient rate depending on speed.				
2	Put robot on test track, run test, and download data.	The robot should travel down the entire length of the test track and then stop.				
3	Plot test data in a spreadsheet program.	Plot of position vs. time should be within 12 – 18 cm 90% of the time.				
	Overall test result:					

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

Test Writer: Sue L. Engineer				
Test Case Name:	Robot integration test #1		**Test ID #:**	Robot-IT-01
Description:	Checks interaction of DC motors on the magnetic compass.		**Type:**	☐ white box ☑ black box
Tester Information				
Name of Tester:			**Date:**	
Hardware Ver:	Robot 1.0		**Time:**	
Setup:	A wooden turn-table should be placed on top of the cardinal direction map. This map should be aligned with a magnetic compass. There should be no metal present while the alignment is being performed. Next, the partially assembled robot should be placed on the turn-table. The MCU should be connected to a terminal to observe and record data.			

Step	Action	Expected Result	Pass	Fail	N/A	Comments
1	Write program to spool compass readings while simultaneously driving motors.	Program should be statically tested to verify accuracy. Should sample compass at a sufficient rate depending on speed.				
2	Run acceptance test	Test program should prompt user to turn the robot to an orientation and then spin the motors will then spin up and down.				
3	Plot spooled data in spreadsheet program.	Plots should be analyzed to see if compass deviated any more than 10 degrees from set point.				
	Overall test result:					

Possibilidades de teste de unidade

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

Unit Test: The Digital Compass

Module	Digital Compass – Geosensor version 2.3
Inputs	- Earth's magnetic field: An orientated field of magnetic force beginning and ending at the earth's magnetic poles. - SClk – Clock signal to clock data through the module. Maximum Frequency is 10Mhz. - SDIn – Serial data input to send data into the compass module. Date is valid on positive SClk edges.
Outputs	- SDOut – Serial data output from the compass module. Data is valid on negative clock edges.
Functionality	Senses the earth's magnetic field and determines the orientation of the compass with respect to the field. This orientation is stored in an internal register and can be retrieved through the SPI interface.
Test	Comp-UT-01

Unit test: compass (matrix)

Test Writer: Sue L. Engineer			
Test Case Name:	Compass unit test #1	**Test ID #:**	Comp-UT-01
Description:	Checks that the compass returns correct angular measurements to the MCU. Test program is in ./test/compass_unit_test_1.c	**Type:**	☐ white box ☑ black box
Tester Information			
Name of Tester:		**Date:**	
Hardware Ver:	Compass Module - Geosensor version 2.3	**Time:**	
Setup:	Compass module should be wired to the MCU through the SPI interface pins. The MCU should be connected to an RS232 terminal through its SCI interface. The terminal should be configured to run at 9600 baud. Cardinal directions map should be aligned using the magnetic compass.		

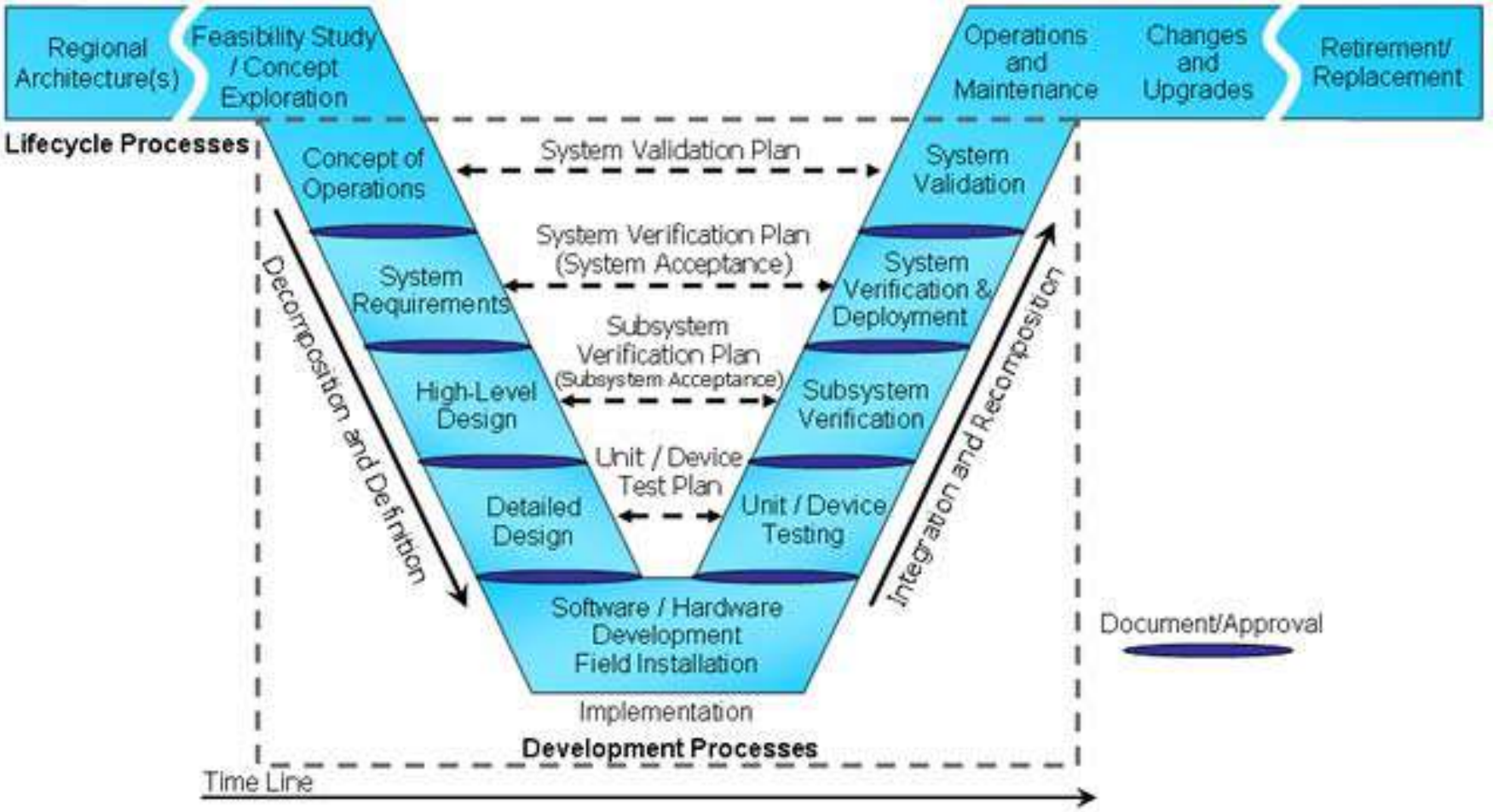
Step	Action	Expected Result	Pass	Fail	N/A	Comments
1	Compile compass.c in /test directory	IDE should generate no warnings or errors.				
2	Download	MCU should report "download successful"				
3	Execute	MCU should display compass splash screen on terminal interface.				
4	Orientate compass to 0 degrees.	Terminal interface should display 0 degrees +/- 10 degrees.				
5	Orientate compass to 30 degrees.	Terminal interface should display 30 degrees +/- 10 degrees.				
6	Orientate compass to 45 degrees.	Terminal interface should display 45 degrees +/- 10 degrees.				
…	…	…				
12	Orientate compass to 315degrees.	Terminal interface should display 315 degrees +/- 10 degrees.				
	Overall test result:					

4 Diretrizes

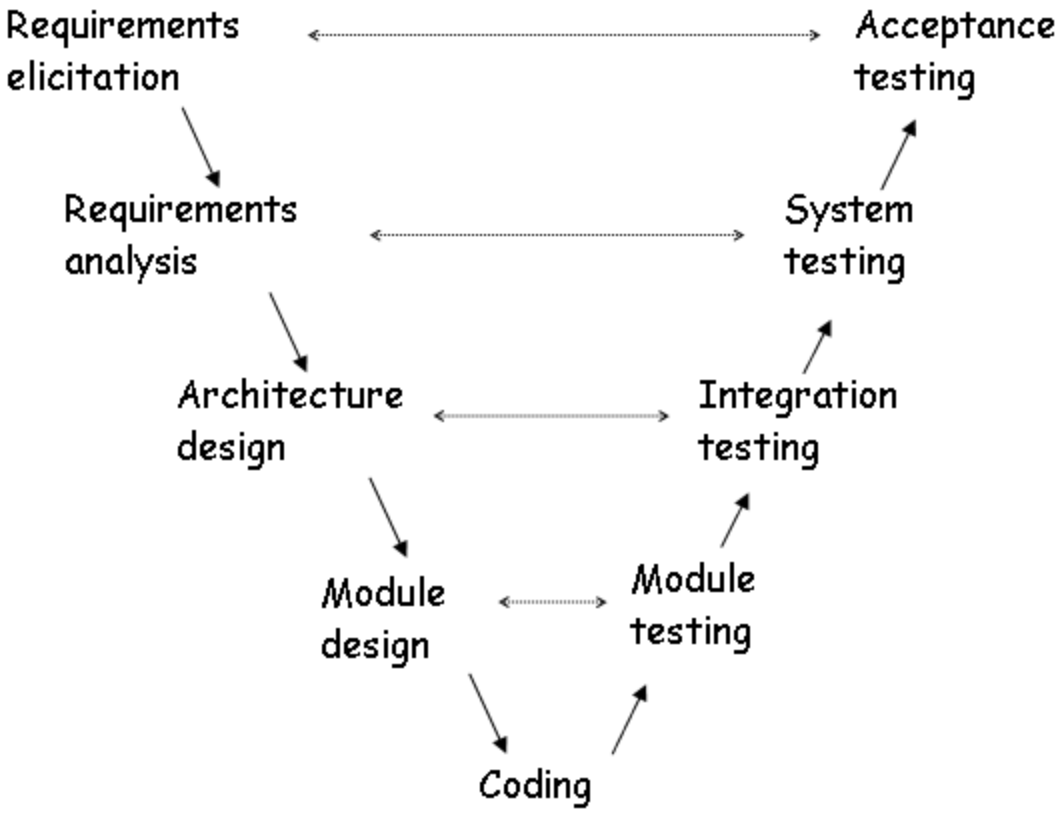
Razões para desenvolver e conduzir testes

- Testes reduzem o numero de bugs.
- Testes constituem-se em boas documentações.
- Testes melhoram o projeto (design).
- Testes permitem a refatoração.
- Testing forçaa voce 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 botto-up do braço direiro do V

- Explique!

Parte 2

Prática