



Curso - Especialização em Engenharia de Software

Disciplina: Métricas de Tamanho, Complexidade e Qualidade e sua utilização em Desenvolvimento de Software

Prof. Edson Saraiva

Aula02 – Garantia da Qualidade de Software

Bibliografia

- Pressman, R., Engenharia de Software, 8ª.ed., McGrawHill, 2001 (Cap 8 – Garantia da Qualidade de Software)
- SOMMERVILLE, I., Engenharia de Software, 8ed., São Paulo: Pearson-Addison Wesley, 2007 (Cap 27 – Gerenciamento da Qualidade)
- SHULL, Forrest J.; BASILI, Victor R. Developing techniques for using software documents: a series of empirical studies. 1998. Tese de Doutorado. research directed by Dept. of Computer Science. University of Maryland, College Park, Md.
- SWBOOK IEEE 2004

Qualidade de Software - introdução

Processo de Desenvolvimento de Software

- Envolve uma série de atividades nas quais, apesar das técnicas, métodos e ferramentas empregados, erros no produto ainda podem ocorrer.

Garantia da Qualidade

- Não é suficiente para uma organização de desenvolvimento de software afirmar que mantém o foco na qualidade na entrega de seus produtos (PRESSMAN, 2006).

PRESSMAN, R., Engenharia de Software, 6ed., São Paulo: McGraw-Hill, 2006 (pag. 577 - panorama)

Qualidade de Software - introdução

Qualidade de Software

- É necessário definir explicitamente o significado de "qualidade de software" no contexto da organização.

Planejar a qualidade

- Planejar um conjunto de atividades que ajudarão a garantir que todo o produto de trabalho de engenharia de software exibe alta qualidade, realizar atividades de controle e garantia da qualidade de software em todo o projeto

PRESSMAN, R., Engenharia de Software, 6ed., São Paulo: McGraw-Hill, 2006 (pag. 577 - panorama)

Valor e Custo da Qualidade

Características de qualidade

- Características de qualidade podem ser requeridas ou não, podem ser requeridas em maior grau ou em menor grau um balanceamento deve ser feito na escolha.

Custo da qualidade

- Pode ser diferenciado entre: custo da prevenção, custo de falhas internas e custo de falhas externas.

Software que agrega valor

- A motivação na aquisição de produtos de software esta relacionada ao desejo de se criar software que tenha valor para o cliente. Uma característica de qualidade pode ser meramente decorativa ou essencial.

Alternativas de qualidade e custo

- O projeto de software deve avaliar as alternativas de qualidade e seu custo

(SWBOOK, 2004)

Modelos e Características de Qualidade

O conceito de qualidade varia entre os modelos

- ISO 9126 - define três modelos de qualidade – qualidade interna, qualidade externa e qualidade em uso.

Qualidade do processo

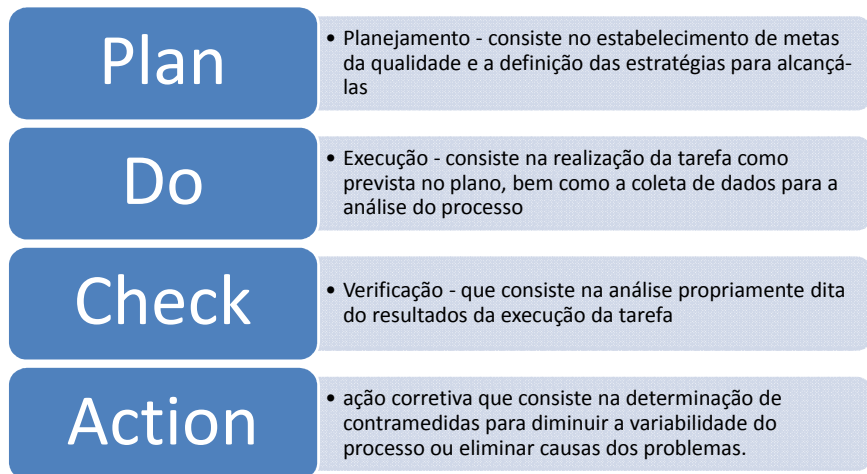
- A qualidade do processo impacta na qualidade do produto, frequentemente não é possível completamente distinguir a qualidade do processo da qualidade do produto. Padrões importantes de qualidade são a ISO 9001, CMMI, MPS.BR.

Qualidade do produto

- A ISO 9126 define um conjunto de características e sub-características de qualidade do produto.

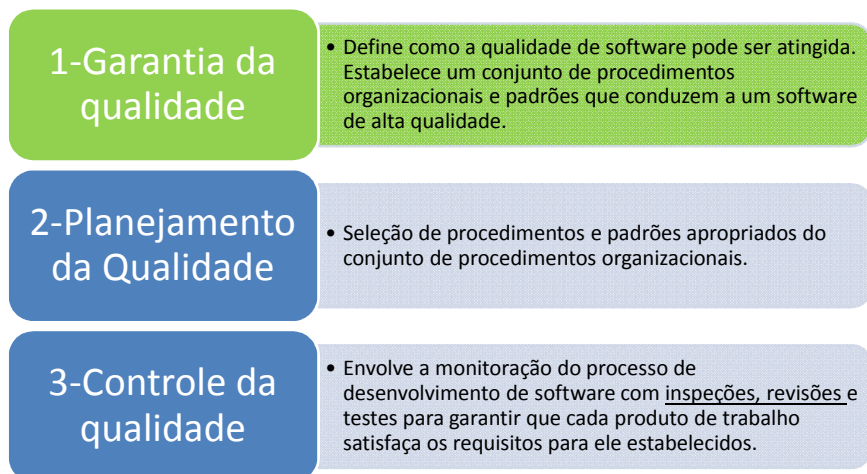
(SWBOOK, 2004)

Melhoria da Qualidade



(SWBOOK, 2004)

Processo de Gerenciamento da Qualidade de Software



SOMMERVILLE, J., Engenharia de Software, 8ed., São Paulo: Pearson-Addison Wesley, 2007 (pag. 424)

1-Garantia da Qualidade

Framework de procedimentos organizacionais

- É o processo de definição de como a qualidade de software pode ser atingida e como a organização de desenvolvimento sabe que o software possui o nível de qualidade necessário. Estabelecimento de um framework de procedimentos organizacionais e padrões que conduzem ao um software de alta qualidade.

Envolve a seleção de padrões

- O processo de GQ está, principalmente, relacionado à definição e seleção de padrões que devem ser aplicados ao processo de desenvolvimento de software e ao produto de software.

Seleção de Ferramentas

- Selecionar e adquirir ferramentas e métodos para apoiar esses padrões

SOMMERVILLE, J., Engenharia de Software, 8ed., São Paulo: Pearson-Addison Wesley, 2007 (pag. 426)

1-Garantia da Qualidade

Padrões de produto

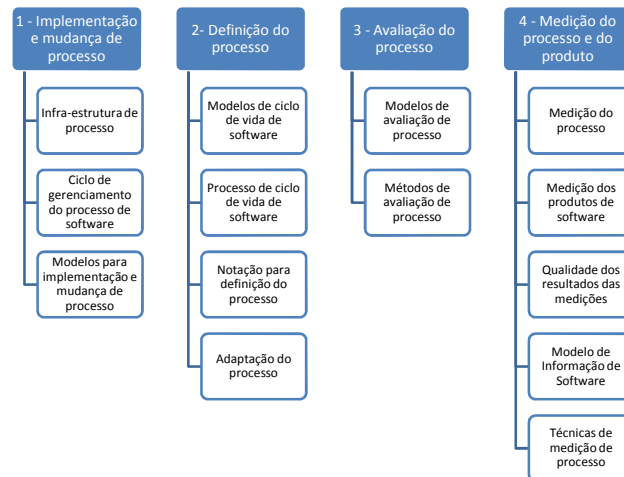
- Padrões de documentos como a estrutura de documentos de requisitos, padrões de codificação como o comentário padronizado para definição de uma classe ou objeto.

Padrões de processo

- Esses padrões definem os processos que devem ser seguidos durante o desenvolvimento de software.

SOMMERVILLE, J., Engenharia de Software, 8ed., São Paulo: Pearson-Addison Wesley, 2007 (pag. 426)

Processo de Engenharia de Software



(SWEBOOK, 2004)

2-Definição do processo

Modelos de ciclo de vida de processo	Processos de ciclo de vida de software	Notação para definição do processo	Adaptação do processo
<ul style="list-style-type: none"> Servem como uma definição de alto nível das fases que ocorrem durante o processo de desenvolvimento Modelo warterfall Modelo evolucionário Modelo espiral Modelo V ESBC 	<ul style="list-style-type: none"> Tendem a ser mais detalhados do que modelos de ciclo de vida de software A principal referencia nesta área é a ISO/IEC 12207 Em algumas situações o processo de engenharia de software deve considerar o contexto organizacional – ISO 9001 (ISO/IEC 90003) Extreme Programming 	<ul style="list-style-type: none"> Processos podem ser definidos em diferentes níveis de abstração IDEFO 	<ul style="list-style-type: none"> Mesmo processos mais detalhados devem ser adaptados para as necessidades locais

(SWEBOOK, 2004)

3-Processo de Avaliação

Modelos de avaliação do processo

- Modelos de avaliação capturam o que é reconhecido como boas práticas
- ISO IEC 15504
- CMMI
- MPS.BR

Métodos de avaliação de processo

- Para executar a avaliação, um método de avaliação necessita ser seguido para quantificar a capacidade do processo ou a maturidade da organização
- CBA-IPI
- SCAMPI

(SWEBOOK, 2004)

4-Medição do processo e do produto

Medição do processo

- Medição é utilizada para identificar forças e fragilidades do processo e avaliar o processo depois de terem sido implementados e/ou modificados. Os recursos primários que precisam ser gerenciados são os recursos humanos. Como resultado as principais medidas de interesse estão relacionadas a produtividade da equipe ou do processo (por exemplo, pontos de função produzidos por recurso)

Medição do processo relacionadas ao produto

- Quantidades de falha por linha de código ou por ponto de função
- Manutenibilidade – esforço para realizar certos tipos de mudanças
- Produtividade – linhas de código por recurso mês, pontos de função por recurso mês.
- Satisfação do cliente

(SWEBOOK, 2004)

4-Medição do processo e do produto

Medição do produto de software

- Tamanho
- Medições estruturais – complexidade ciclomatica
- Medidas de qualidade ISO 9126

Modelos de informação de software

- Como os dados são coletados e armazenados em um repositório é possível construir modelos usando dados e conhecimento.
- Estes modelos existem para propósitos de análise, classificação e previsão.

Técnicas de medição de processo

- Técnicas de medição podem ser usadas para identificar forças e fragilidades no processo de engenharia de software.
- Técnicas analíticas
- Benchmarking

(SWEBOOK, 2004)

4-Medição do processo e do produto

Técnicas analíticas

- Estabelecem evidências quantitativas para determinar onde as melhorias são necessárias e se uma iniciativa de melhoria obteve sucesso.
- Estudo experimental-um novo processo é comparado com o processo atual para determinar se houve melhora
- Revisão do processo – o processo é revisado e potenciais melhorias no processo são identificadas, os indivíduos que executam a análise do processo usam seus conhecimentos e competências para decidir se uma modificação potencialmente levará a um resultado desejado. A observação pode oferecer um feedback útil para identificar melhorias no processo.
- Classificação ortogonal de defeitos
- Análise de causa raiz
- CEP

Benchmarking

- Supõe que uma organização com menos maturidade adote práticas de uma organização com alta maturidade e consequentemente também obtenha o mesmo nível de maturidade.

(SWEBOOK, 2004)

Processo de Gerenciamento da Qualidade de Software

1-Garantia da qualidade

- Define como a qualidade de software pode ser atingida. Estabelece um conjunto de procedimentos organizacionais e padrões que conduzem a um software de alta qualidade.

2-Planejamento da Qualidade

- Seleção de procedimentos e padrões apropriados do conjunto de procedimentos organizacionais.

3-Controle da qualidade

- Envolve a monitoração do processo de desenvolvimento de software com inspeções, revisões e testes para garantir que cada produto de trabalho satisfaça os requisitos para ele estabelecidos.

SOMMERVILLE, J., Engenharia de Software, 8ed., São Paulo: Pearson-Addison Wesley, 2007 (pag. 424)

2-Planejamento da Qualidade

1-Apresentação do produto

- Descrição do produto, o mercado pretendido e as expectativas de qualidade para o produto.

2-Descrições de processo

- Os processos de desenvolvimento de software e de serviços devem ser usados para o desenvolvimento e gerenciamento do produto

3-Plano de produto

- Datas críticas de liberação e as responsabilidades para o produto junto com planos para serviços de distribuição e de produto.

4-Metas de qualidade

- As metas de qualidade para o produto incluindo identificação e justificativa de atributos críticos de qualidade de produto

5-Riscos e gerenciamento de riscos

- Os riscos principais que poderiam afetar a qualidade de produto e as ações para tratar esses riscos.

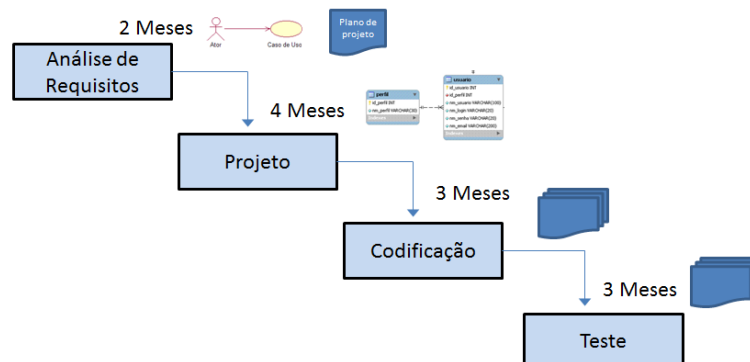
SOMMERVILLE, J., Engenharia de Software, 8ed., São Paulo: Pearson-Addison Wesley, 2007 (pag. 430)

2-Planejamento da Qualidade

1. Apresentação do produto
2. Descrição do processo – envolve estabelecer uma série de marcos
 - Um marco é um ponto final reconhecível de uma atividade do processo de desenvolvimento do software. A cada marco, deve existir uma saída formal, como um relatório, que possa ser apresentado a gerência. Os marcos devem representar o fim de um estágio lógico e distinto do projeto.
 - Para estabelecer os marcos, o processo de software deve ser decomposto em atividades básicas com saídas associadas.

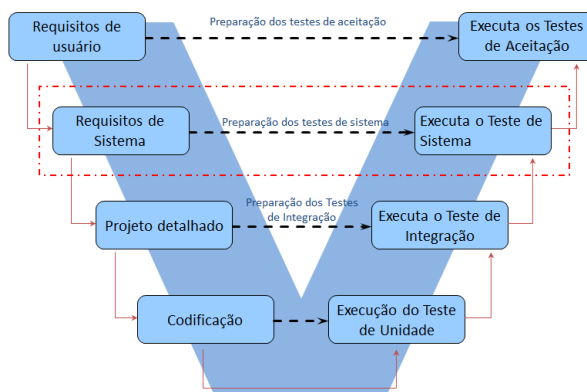
2-Planejamento da Qualidade

2. Descrição do processo - modelo waterfall



2-Planejamento da Qualidade

2. Descrição do processo - modelo V



CRAIG, Rick David; JASKIEL, Stefan P.; JASKIEL, Stefan P. **Systematic software testing**. Boston: Artech House, 2002.

2-Planejamento da Qualidade

2. Descrição do processo - modelo em espiral

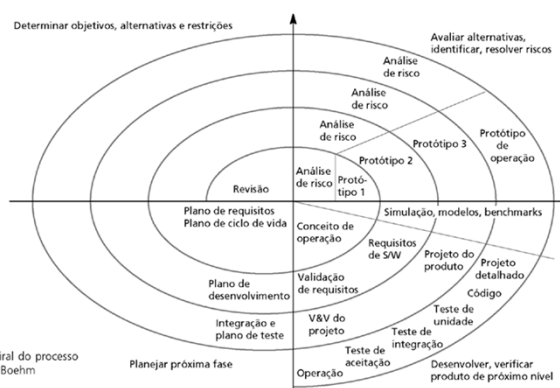


Figura 4.5

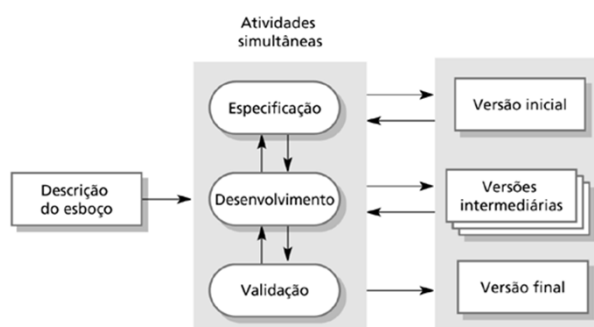
Modelo em espiral do processo de software de Boehm (©IEEE, 1988).

2-Planejamento da Qualidade

2. Descrição do processo - modelo evolucionário

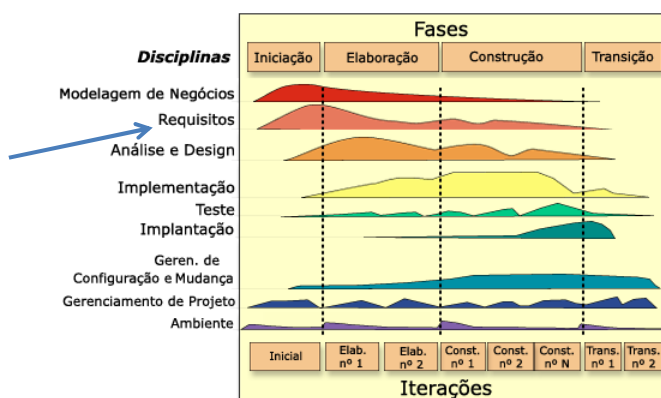
Figura 4.2

Desenvolvimento evolucionário.



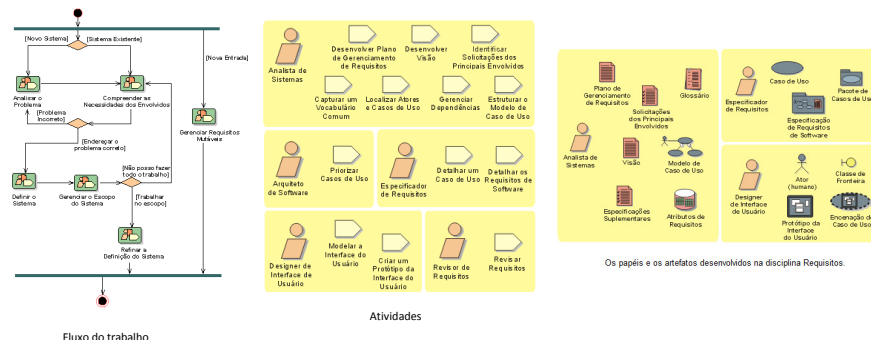
2-Planejamento da Qualidade

2. Descrição do processo - RUP



2-Planejamento da Qualidade

2. Descrição do processo – RUP disciplina requisitos



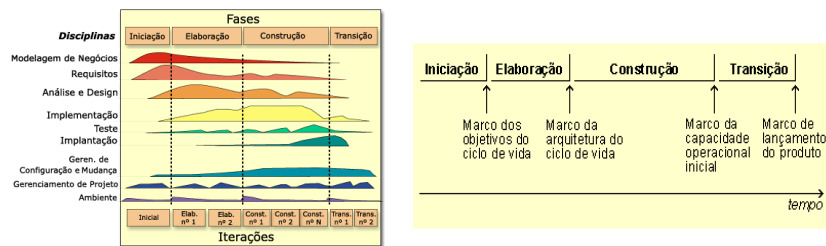
2-Planejamento da Qualidade

2. Descrição do processo – envolve estabelecer uma série de marcos

- Um marco é um ponto final reconhecível de uma atividade do processo de desenvolvimento do software. A cada marco, deve existir uma saída formal, como um relatório, que possa ser apresentado a gerência. Os marcos devem representar o fim de um estágio lógico e distinto do projeto.
- Para estabelecer os marcos, o processo de software deve ser decomposto em atividades básicas com saídas associadas.

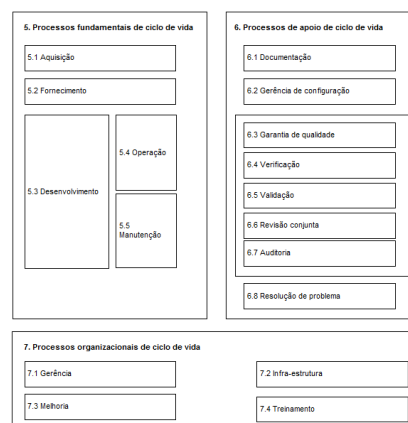
2-Descrever o modelo de processo

- Marcos - RUP



4-Metas de Qualidade - processo

- Aderência a norma ISO/IEC 12207



4-Metas de Qualidade - processo

- Aderência a norma ISO/IEC 12207
- 5.3 Processo de desenvolvimento
 - O processo de desenvolvimento contém as atividades e tarefas do desenvolvedor.
 - O processo contém as atividades para análise de requisitos, projeto, codificação, integração, testes, instalação e aceitação relacionada aos produtos de *software*.
 - Pode conter atividades relacionadas ao sistema, se estipulado no contrato. O desenvolvedor executa ou apoia as atividades neste processo, de acordo com o contrato.

5.3 Processo de desenvolvimento

Lista das atividades. Este processo consiste nas seguintes atividades:

- 1) Implementação do processo;
- 2) Análise dos requisitos do sistema;
- 3) Projeto da arquitetura do sistema;
- 4) Análise dos requisitos do *software*;
- 5) Projeto da arquitetura do *software*;
- 6) Projeto detalhado do *software*;
- 7) Codificação e testes do *software*;
- 8) Integração do *software*;
- 9) Teste de qualificação do *software*
- 10) Integração do sistema;
- 11) Teste de qualificação do sistema;
- 12) Instalação do *software*;
- 13) Apoio à aceitação do *software*.

5.3 Processo de desenvolvimento

- **5.3.1 Implementação do processo.** Esta atividade consiste na seguinte tarefa:
- **5.3.1.1** Se não estiver estipulado no contrato, o desenvolvedor deve definir ou selecionar um modelo de ciclo de vida de *software* apropriado ao escopo, magnitude e complexidade do projeto. As atividades e tarefas do processo de desenvolvimento devem ser selecionadas e mapeadas no modelo de ciclo de vida.

5.3 Processo de desenvolvimento

- **5.3.2 Análise dos requisitos do sistema.** Esta atividade consiste nas seguintes tarefas, as quais o desenvolvedor deve executar ou apoiar conforme especificado no contrato:
- **5.3.2.1** O uso específico pretendido do sistema a ser desenvolvido deve ser analisado para especificar os requisitos do sistema. A especificação dos requisitos do sistema deve descrever: funções e capacidades do sistema; requisitos de negócios, organizacionais e de usuários; requisitos de proteção, de segurança, de engenharia de fatores humanos (ergonomia), de interface, de operações e de manutenção; restrições de projeto e requisitos de qualificação. A especificação dos requisitos do sistema deve ser documentada.
- **5.3.2.2** Os requisitos do sistema devem ser avaliados, considerando os critérios listados a seguir. Os resultados das avaliações devem ser documentados.
 - Rastreabilidade para as necessidades de aquisição;
 - Consistência com as necessidades de aquisição;
 - Testabilidade;
 - Viabilidade do projeto da arquitetura do sistema;
 - Viabilidade da operação e manutenção.

5.3 Processo de desenvolvimento

- **5.3.3 Projeto da arquitetura do sistema.** Esta atividade consiste nas seguintes tarefas, as quais o desenvolvedor deve executar ou apoiar conforme especificado no contrato:
 - **5.3.3.1** Uma arquitetura de alto nível do sistema deve ser estabelecida. A arquitetura deve identificar itens de *hardware*, *software* e operações manuais. Deve ser assegurado que todos os requisitos do sistema sejam alocados entre os itens. Itens de configuração de *hardware*, itens de configuração de *software* e operações manuais devem ser subsequentemente identificados, a partir destes itens. A arquitetura do sistema e os requisitos do sistema alocados aos itens devem ser documentados.
 - **5.3.3.2** A arquitetura do sistema e os requisitos para os itens devem ser avaliados, considerando os critérios listados a seguir. Os resultados das avaliações devem ser documentados.
 - Rastreabilidade para os requisitos do sistema;
 - Consistência com os requisitos do sistema;
 - Adequação dos métodos e padrões de projeto utilizados;
 - Viabilidade de os itens de *software* atenderem seus requisitos alocados;
 - Viabilidade da operação e da manutenção.

4-Metas de qualidade - produto

Potencial de exposição a falhas

- Análise de risco e priorização, os ct são selecionados baseados na cobertura das funções de maior risco.

Para cada requisito determinar

- Probabilidade de falha (1-baixa, 2- média, 3-alta)
- Consequência (impacto da falha) (1-baixa, 2-médio, 3-alta)
- Coeficiente de risco = probabilidade X consequência (impacto da falha)

Análise do perfil operacional

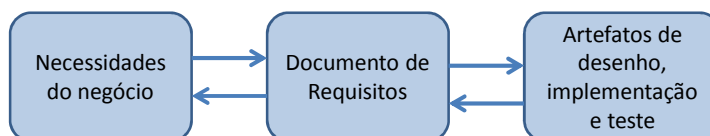
- Funções mais utilizadas

Kaner, C., Blackbox Software Testing: risk based testing, Fall 2005

4-Metas de qualidade - produto

Rastreabilidade

- Pode ser implementada por um conjunto de ligações entre requisitos inter-relacionados, entre requisitos e suas fontes e entre requisitos e os componentes que os implementam



4-Metas de qualidade - produto

- Uma organização desenvolvedora de software pode usar medições para planejar e controlar a qualidade do produto durante o desenvolvimento, avaliar a qualidade do produto final e aprender sobre o processo de software e sobre o produto.

Características de qualidade		Atributo	Métrica	Limite superior
Manutenibilidade	Corretiva	Tempo de recuperação	Número médio de horas necessárias para recuperação do sistema (produção parada)	1h
		Coesão	Número médio de módulos modificados para corrigir o erro	1,2

Processo de Gerenciamento da Qualidade de Software

1-Garantia da qualidade

- Define como a qualidade de software pode ser atingida. Estabelece um conjunto de procedimentos organizacionais e padrões que conduzem a um software de alta qualidade.

2-Planejamento da Qualidade

- Seleção de procedimentos e padrões apropriados do conjunto de procedimentos organizacionais.

3-Controle da qualidade

- Envolve a monitoração do processo de desenvolvimento de software com inspeções, revisões e testes para garantir que cada produto de trabalho satisfaça os requisitos para ele estabelecidos.

SOMMERVILLE, J., Engenharia de Software, 8ed., São Paulo: Pearson-Addison Wesley, 2007 (pag. 424)

3-Controle de Qualidade

Controle da qualidade

- Envolve a monitoração do processo de desenvolvimento de software (revisões, inspeções e testes) para assegurar que os procedimentos e os padrões de garantia da qualidade estão sendo seguidos.

Revisões de qualidade

- Processos e documentação usados para produção de software são revisados por um grupo de pessoas.

Avaliação automatizada

- O software e os documentos produzidos são processados por algum programa e comparados com padrões que se aplicam ao projeto específico.

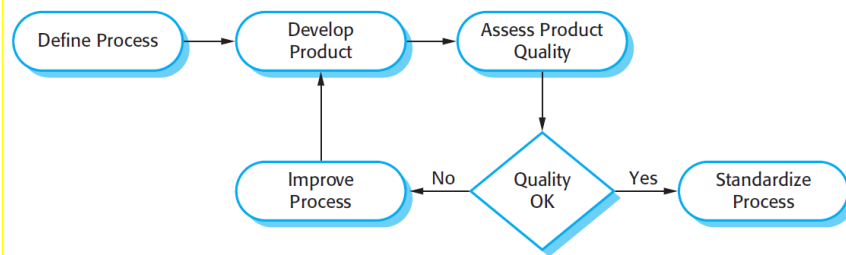
Ferramentas para acelerar o processo de revisão

- A medição de software se dedica a derivar um valor numérico para algum atributo de um produto de software ou de um processo comparando esses valores com os padrões que se aplicam a uma organização pode-se obter conclusões sobre a qualidade do software ou dos processos de software.

SOMMERVILLE, J., Engenharia de Software, 8ed., São Paulo: Pearson-Addison Wesley, 2007 (pag. 431)

3-Controle de Qualidade

- A suposição fundamental da gerência de qualidade é que a qualidade do processo de desenvolvimento afeta diretamente a qualidade dos produtos entregues. Deve-se medir a qualidade do produto e mudar o processo até o nível de qualidade que a organização deseja atingir.



Qualidade baseada em processo

SOMMERVILLE, J., Engenharia de Software, 8ed., São Paulo: Pearson-Addison Wesley, 2007 (pag. 425)

Considerações Práticas

O processo de gerenciamento da qualidade encontra defeitos

- Caracterizar estes defeitos facilita a correção do processo ou do produto e informa o gerente do projeto ou o cliente do status do processo ou produto.

Taxonomia de defeitos

- Permite classificar os defeitos

Falhas encontradas na atividade de teste

- Falhas detectadas no ambiente interno e externo

Medições

- Os modelos de qualidade do produto de software frequentemente incluem medições para determinar o grau de qualidade de cada característica de qualidade

Taxonomia de defeitos

- Identificação de defeitos em documento de requisitos

Omissão	Qualquer requisito significante relacionado a funcionalidade, desempenho, restrições de projeto, atributos, ou interface externa que tenha sido omitido do documento de requisitos.
Informação ambígua	Informação no documento de requisitos que pode ser interpretada de várias maneiras.
Informação inconsistente	Dois ou mais requisitos que se contradizem.
Fato incorreto	Algum fato declarado no documento de requisitos que não pode ser verdade diante das condições especificadas pelo sistema.
Informação Extra	Informação desnecessária ou não usada.
Defeitos diversos	Outros defeitos, tais como incluir um requisito na seção errada do documento de requisitos.

SHULL, Forrest J.; BASILI, Victor R. Developing techniques for using software documents: a series of empirical studies. 1998. Tese de Doutorado. research directed by Dept. of Computer Science. University of Maryland, College Park, Md.

Taxonomia de defeitos

- Identificação de defeitos em documento de requisitos

- **Omissão:** um defeito do tipo omissão é caracterizado quando: (1) algum requisito importante relacionado à funcionalidade, ao desempenho, às restrições de projeto, ao atributo, ou à interface externa não foi incluído; (2) não está definida a resposta do software para todas as possíveis situações de entrada de dados; (3) faltam seções no documento de requisitos; (4) faltam referências de figuras, tabelas, e diagramas; (5) falta definição de termos e unidades de medidas.
- **Ambigüidade:** um requisito tem várias interpretações devido a diferentes termos utilizados para uma mesma característica ou vários significados de um termo para um contexto em particular.
- **Inconsistência:** dois ou mais requisitos são conflitantes e o conhecimento sobre o domínio não permite identificar qual requisito é verdadeiro.
- **Fato Incorreto:** um requisito descreve um fato que não é verdadeiro, considerando-se as condições solicitadas para o sistema.
- **Informação Estranha:** as informações fornecidas no requisito não são necessárias ou mesmo usadas.
- **Outros:** outros defeitos como a inclusão de um requisito numa seção errada do documento.

Taxonomia de defeitos

- Identificação de defeitos em documentos arquiteturais

Tipos de defeitos	Descrição
Omissão	1. Quando um elemento arquitetural necessário para o atendimento a um requisito não foi definido;
Ambigüidade	2. Quando a forma como os elementos arquiteturais ou suas responsabilidades foram definidos dificulta ou impossibilita o atendimento a um requisito de qualidade. 3. Quando elementos descritos em visões distintas possuem o mesmo nome, mas responsabilidades diferentes (Homônimo);
Inconsistência	4. Quando elementos descritos em visões distintas possuem mesma responsabilidade, mas nomes distintos (Sinônimo); 5. Quando um elemento arquitetural presente em diagramas das demais visões não foi definido no diagrama avaliado; 6. Quando a representação não condiz com a semântica estabelecida pela abordagem de documentação. 7. Quando um elemento arquitetural é definido com responsabilidades distintas em duas ou mais visões. 8. Quando um elemento é representado de maneira diferente em duas visões.
Fato Incorreto	9. Quando um elemento não foi descrito ou representado de forma correta 10. Quando não é possível mapear um elemento arquitetural para algum elemento descrito em outra visão.
Informação Estranha	11. Quando não é possível determinar o papel de um elemento arquitetural ou de uma de suas responsabilidades no atendimento aos requisitos especificados.