



Escola Superior de Tecnologia e Gestão  
Politécnico do Porto

**Author**

Cristóvão Sousa

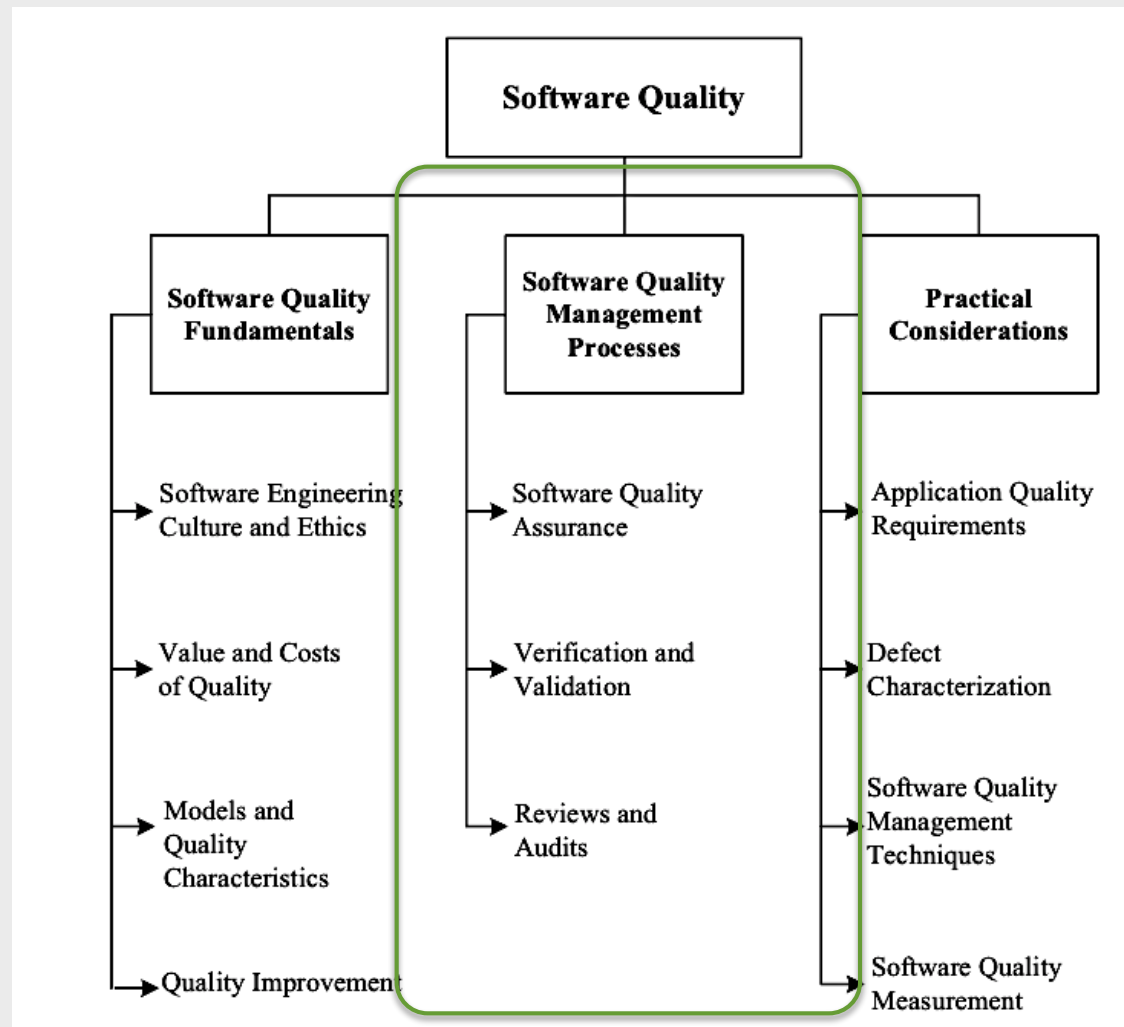
**Version**

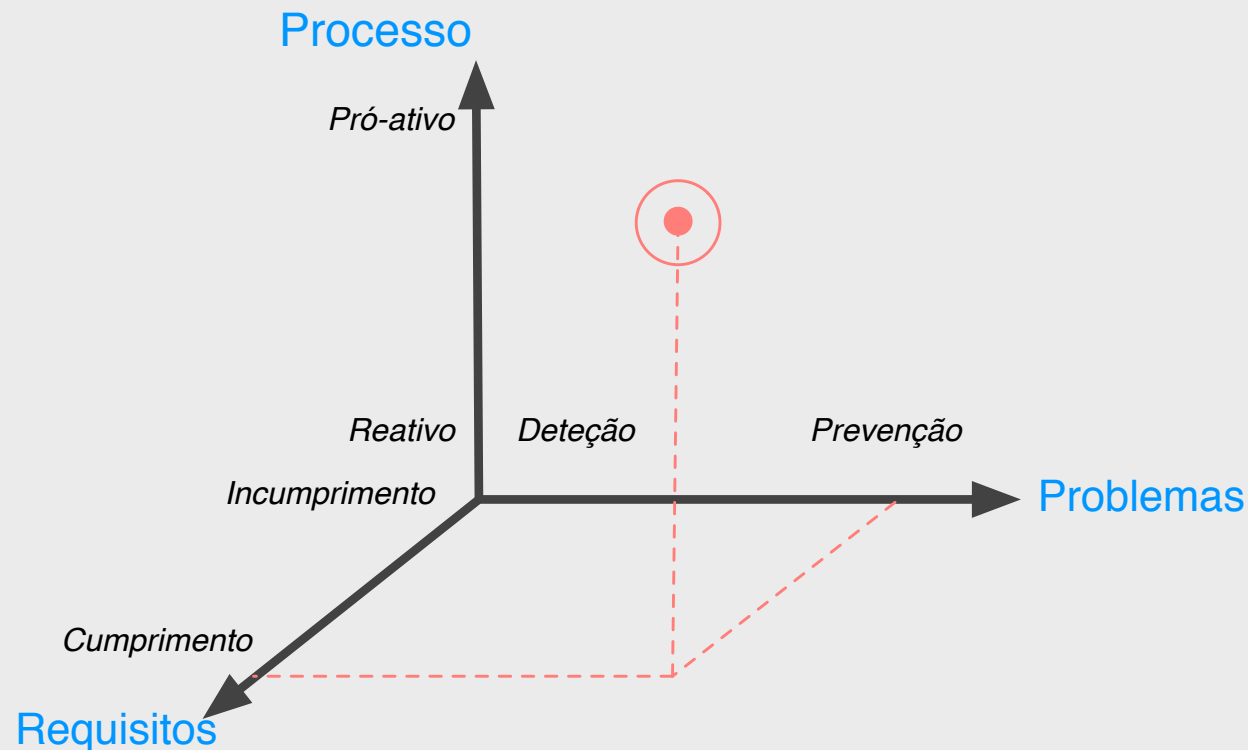
v1.0

# Gestão da Qualidade de Software

***Software Quality Assurance***

**Summary | Quality Management • SQA • Verification and Validation Processes**





- Como Desenvolver software de elevada qualidade?

- **Define:**
  - **processos;**
  - **donos dos processos e os requisitos desses processos;**
  - **medidas dos processos** e os seus outputs e canais de feedback
- Consiste em várias actividades e **envolve**: a **definição** do produto no que diz respeito às suas **características de qualidade; Planear todo o processo** para alcançar o referido produto.

“The software quality management processes must address how well software products will, or do, satisfy customer and stakeholder requirements, provide value to the customers and other stakeholders, and provide the software quality needed to meet software requirements.”

[Swebok, 2004]

- Alguns dos processos especificados pela Gestão da Qualidade do Software encontram-se definidos no Standard (IEEE12207.0-96), nomeadamente:
  - **Quality assurance process** (👉)
  - **Verification process** (✓)
  - **Validation process** (✓)
  - Review process



- A Garantia da Qualidade do Software é “um **conjunto de acções planeadas** e sistemáticas, necessárias **para assegurar a Qualidade no Software** ” [Schulmeyer, 1987]
- A Garantia da Qualidade do Software é composta por várias tarefas associadas a dois grupos diferentes:
  - **Engenheiros de Software**, que executam o trabalho técnico
  - A **Equipa de SQA** (Software Quality Assurance), responsável pelo planeamento, acompanhamento, registo de informação, análise e relatórios sobre a Garantia da Qualidade

*É possível controlar o que  
não se pode medir?*

*Pode-se medir o que não é  
possível controlar?*



“SQA processes provide assurance that the software products and processes in the project life cycle conform to their specified requirements by **planning**, enacting, and performing a set of activities to provide adequate confidence that quality is being built into the software”

[Swebok, 2004]

- O plano SQA define os meios que irão ser usados para garantir que o software desenvolvido no âmbito de um determinado produto de software satisfaz os requisitos do utilizador e que respeita amplamente as restrições do projecto a um nível de qualidade alto.
- Como?
  - primeiro é necessário definir e perceber claramente o “quality target”.
  - deve ser considerado: a gestão, o desenvolvimento e os planos de manutenção para o software.
  - O standard IEEE730-98, define os detalhes do “Software Quality Assurance Plan”

- O que contém o SQA Plan
  - identifica:
    - documentos, standards, práticas e convenções que regem o projecto de modo a controlar e monitorar para garantir a adequação e conformidade;
    - medidas, técnicas (estatísticas) e procedimento para reportar problemas e acções correctivas, recursos (ferramentas, técnicas e metodologias), formação e indicações de como criar documentação específica

- contempla também:
  - outras actividades para garantia de qualidade de software que estejam no SQA Plan, nomeadamente:
  - procurement of supplier software (aquisição de software) ou
  - commercial off-the-shelf software (COTS), e
  - serviços pós-venda;
- pode ainda conter:
  - critérios de aceitação e ainda actividades de report e gestão consideradas críticas para a qualidade de software

COTS - software comercial e específico

# Categories of SQA work

- Um trabalho típico de garantia da qualidade de software abrange 6 dimensões:
  - métodos e ferramentas de construção
  - revisões formais
  - estratégia de teste \*
  - controlo de documentação e histórico de mudanças
  - procedimentos para garantir a adequação aos padrões de desenvolvimento
  - mecanismos de medição e análise \*



\* Erradamente, estas duas dimensões são pouco valorizadas.

# Categories of SQA work



- No caso dos testes
  - O erro começa em desvalorizar o processo de testes
  - Existem ferramentas de teste que permitem a automação dos testes, no entanto, deve-se usar a mais adequada e ter um processo de testes claro e definido
  - Efectuar planeamento dos testes
  - O custo do processo e de uma ferramenta de testes será diluído no tempo e nos “bugs” eliminados previamente.
- No caso dos mecanismos de medição
  - o problema é a indefinição do que se deve medir
  - o erro é desenvolver o software de forma reactiva e não de forma pró-activa, corrige-se aquilo que é reclamado ou questionado, se ninguém analisou não tem erros.....

Os testes e mecanismos de medição complementam, mas não solucionam todo o processo de construção de software, pois em todas as dimensões existe o factor humano no processo de construção que é um factor chave do sucesso.

# Software Quality - direct and indirect factors



- Existem dois grandes grupos que afectam a qualidade de software:

- o que pode ser **medido de forma directa**

- ex.: número de erros, linhas de código, etc.

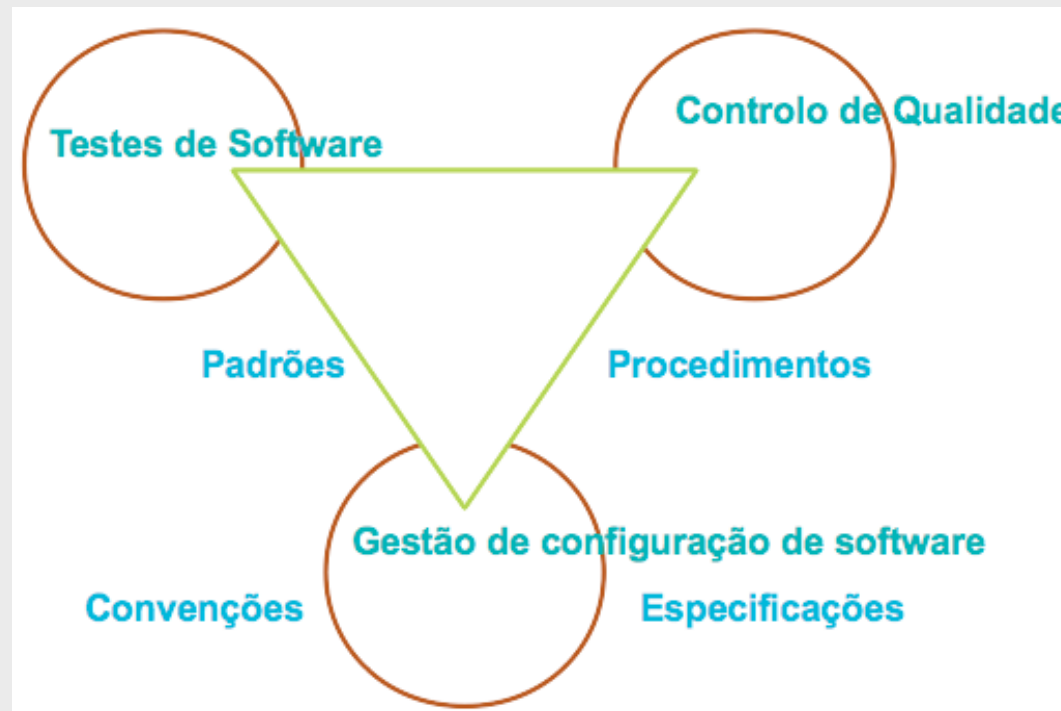


Para cada situação  
deve haver um  
processo de medição

- o que pode ser **medido de forma indirecta**

- ex.: usabilidade, etc.

# SQA main components / subprocesses







- Estratégia mais popular de gestão de risco
- Usados para verificar o encontro dos requisitos com o produto
- Nem todos os defeitos são descobertos durante os testes
- Testes de software incluem actividades de verificação e validação das actividades do processo de desenvolvimento

- Processos e métodos usados para monitorar o trabalho e os requisitos envolvidos
- É focado nas revisões e remoção de defeitos antes da entrega do produto
- Consiste em checks do produto bem definidos que sejam especificados dentro do plano de garantia de qualidade
- Revisões de especificação, inspecções de código e documentos, e checks de entrega ao utilizador



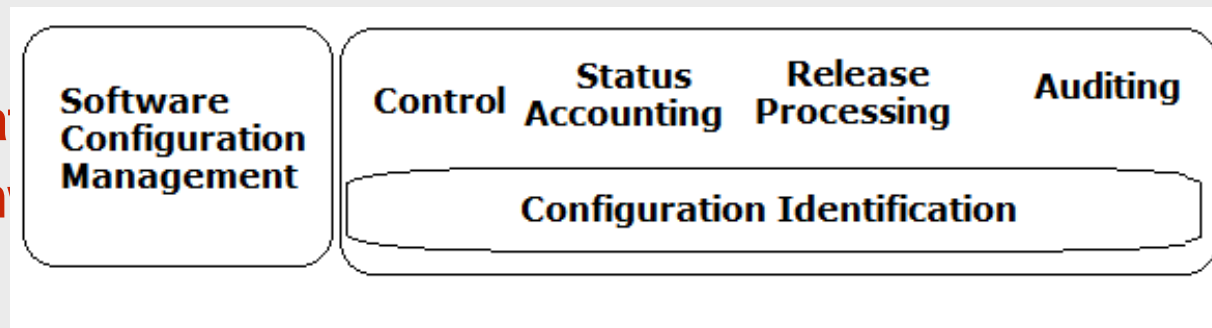
# SQA - Software Config. Management



- Refere-se a “etiquetar”, rastrear e controlar as mudanças nos elementos do software ou do sistema
- Controla a evolução do sistema pela gestão das versões dos componentes de software e seus relacionamentos
- O objectivo é identificar todos os componentes do software inter-relacionados e controlar a sua evolução através das fases do ciclo de vida do software
- Controla o código e documentos associados de tal maneira que o código final e a sua descrição sejam consistentes e representem aqueles itens que eventualmente foram revistos e testados

- Gestão de configuração de SW (engloba fundamentalmente):
  - component identification;
  - version control;
  - config. building;
  - change control;

Conjunto de atividades do ciclo de vida de desenvolvimento



do ciclo de

## NECESSIDADE

- Porque trabalho em equipa;
- Porque há diferentes versões de software;
- Porque há várias *releases* para o cliente;

## Logo,

- Há a necessidade de gerir o processo e garantir “agilidade” no mesmo!

## TAREFAS

- Identificar *work products (WP)* susceptíveis mudança;
- Estabelecer relações entre eles (os WP);
- Estabelecer mecanismos de gestão das diferentes versões desses WP
- Controlar as alterações
- Auditar e reportar as alterações

## O QUE NA PRÁTICA SIGNIFICA

- Configuration Item
- Baseline
- Versão
- Release



## CONFIGURATIONS ITEMS

- hardware e software que necessitam controlo
  - Documento de requisitos
  - Documentos com o desenho da solução
  - Documentos de teste (plano de teste)
  - Código fonte
  - Executáveis, scripts, ...
  - Dados/informação de testes
  - Bugs
  - Build
  - Servers

## VERSÃO

- Versão específica de um determinado *configuration item*

## BASELINE

- Especificação de um produto que foi formalmente revista e acordada com o responsável da gestão de configurações e que serve de base para desenvolvimento futuros e que evolui de acordo com o controlo de versões
- À medida que o o produto é desenvolvido, poderão existir diferentes baselines
  - Development Baseline
  - Product Baseline
  - Functional Baseline

## RELEASE

- Distribuição formal de uma versão aprovada

## PROCESSO

- **Identificação**

- Identificar e designar (dar um nome) a cada item;
- Categorizar cada item;

- **Controlo de alterações**

- Pedido de alterações > Aprovação/Rejeição > Checkout > Executar as alterações > Rever

- **Controlo de versões**

## PROCESSO

- **Auditorias às configurações**

- Garantir o “alinhamento” com o plano de configurações
- Identificar qualquer não-conformidade
- Aplicar medidas corretivas

- **Reporting**

- Status da auditoria
- Quem, Quando e o Quê?

## Contents

1. Overview .....	1
1.1 Scope .....	1
2. References .....	1
3. Definitions and acronyms .....	3
3.1 Definitions .....	3
3.2 Acronyms .....	3
4. Software Quality Assurance Plan.....	4
4.1 Purpose (Section 1 of the SQAP) .....	4
4.2 Reference documents (Section 2 of the SQAP) .....	4
4.3 Management (Section 3 of the SQAP) .....	5
4.4 Documentation (Section 4 of the SQAP) .....	5
4.5 Standards, practices, conventions, and metrics (Section 5 of the SQAP) .....	7
4.6 Reviews and audits (Section 6 of the SQAP) .....	7
4.7 Test (Section 7 of the SQAP) .....	9
4.8 Problem reporting and corrective action (Section 8 of the SQAP) .....	9
4.9 Tools, techniques, and methodologies (Section 9 of the SQAP) .....	9
4.10 Code control (Section 10 of the SQAP) .....	9
4.11 Media control (Section 11 of the SQAP) .....	9
4.12 Supplier control (Section 12 of the SQAP) .....	9
4.13 Records collection, maintenance, and retention (Section 13 of the SQAP) .....	10
4.14 Training (Section 14 of the SQAP) .....	10
4.15 Risk management (Section 15 of the SQAP) .....	10