



Gestão de configurações

Metodologias e Desenvolvimento de Software

Pedro Salgueiro

pds@di.uevora.pt

CLV-256



- Outline
 - Gestão de alterações
 - Gestão de versões
 - Construção do sistema



Gestão de configurações

- Software muda frequentemente.
 - Sistemas podem ser vistos como uma série de versões
 - Cada versão tem de ser mantida e gerida
- Cada versão
 - Implementa pedidos de alterações
 - Correções de falhas
 - Adaptações diversas
- Gestão de configurações
 - “Políticas” e processos para gerir alterações/mudanças
 - Essencial
 - Muito fácil de perder o rasto das alterações e de quais as alterações que devem fazer parte de uma versão do sistema

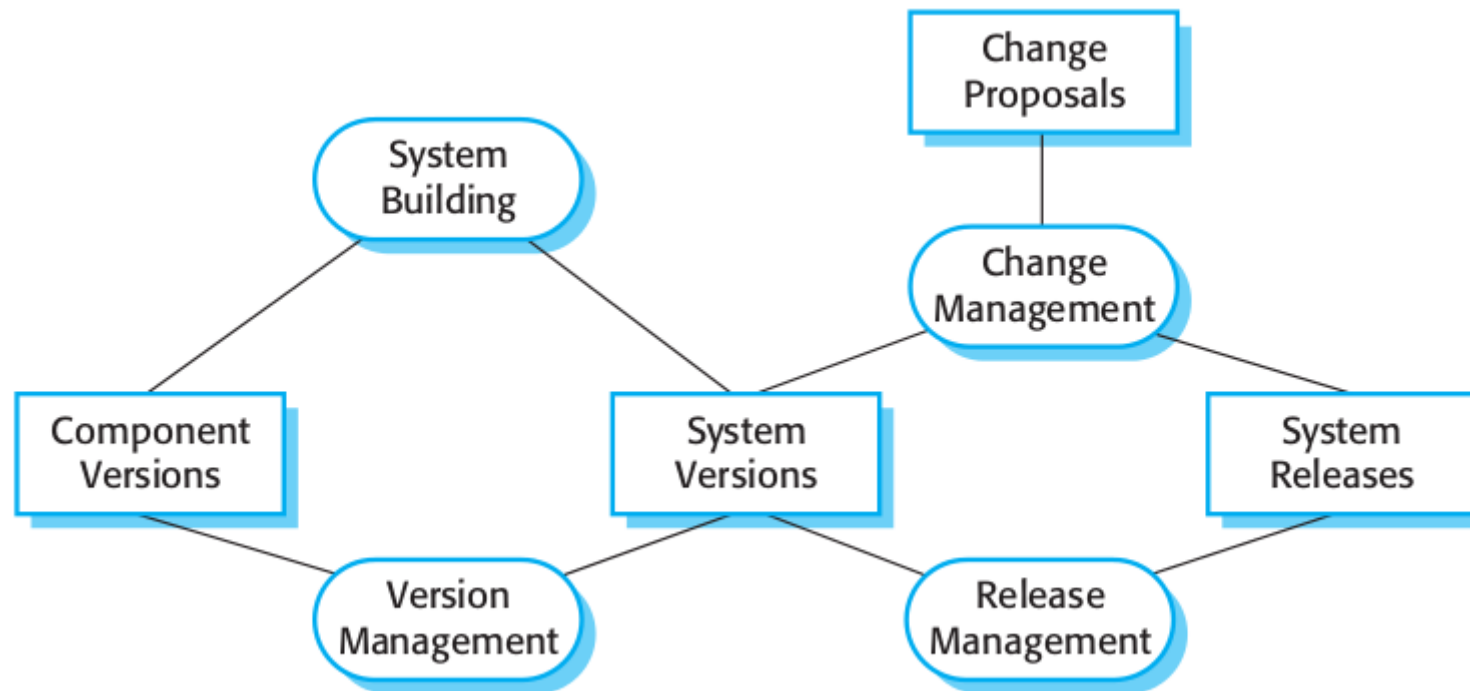


Gestão de configurações - actividades

- Gestão de alterações
 - Manter o “rasto” dos pedidos de alterações feitas pelos clientes e programadores, quais os custos dos impactos destas alterações e decidir quais as alterações que devem ser implementadas.
- Gestão de versões
 - Manter o rasto das várias versões dos componentes do sistema, garantindo que as alterações dos diferentes programadores não interfere com outras.
- Criação do sistema (system building)
 - Processo de construir componentes do sistema, dados e bibliotecas, para depois compilar tudo num sistema executável
- Gestão de *releases*
 - Preparar o software para ser entregue e manter registo de todas as versões entregues (ao cliente).



Gestão de configurações - actividades





Gestão de configurações - terminologia

Term	Explanation
Configuration item or software configuration item (SCI)	Anything associated with a software project (design, code, test data, document, etc.) that has been placed under configuration control. There are often different versions of a configuration item. Configuration items have a unique name.
Configuration control	The process of ensuring that versions of systems and components are recorded and maintained so that changes are managed and all versions of components are identified and stored for the lifetime of the system.
Version	An instance of a configuration item that differs, in some way, from other instances of that item. Versions always have a unique identifier, which is often composed of the configuration item name plus a version number.
Baseline	A baseline is a collection of component versions that make up a system. Baselines are controlled, which means that the versions of the components making up the system cannot be changed. This means that it should always be possible to recreate a baseline from its constituent components.
Codeline	A codeline is a set of versions of a software component and other configuration items on which that component depends.



Gestão de configurações - terminologia

Term	Explanation
Mainline	A sequence of baselines representing different versions of a system.
Release	A version of a system that has been released to customers (or other users in an organization) for use.
Workspace	A private work area where software can be modified without affecting other developers who may be using or modifying that software.
Branching	The creation of a new codeline from a version in an existing codeline. The new codeline and the existing codeline may then develop independently.
Merging	The creation of a new version of a software component by merging separate versions in different codelines. These codelines may have been created by a previous branch of one of the codelines involved.
System building	The creation of an executable system version by compiling and linking the appropriate versions of the components and libraries making up the system.



Gestão de alterações

- Necessidades organizacionais e requisitos mudam durante a vida de um sistema, problemas têm de ser corrigidos e os sistemas têm de se adaptar a alterações no ambiente de funcionamento
- Gestão de alterações tem como objectivo garantir que a evolução de um sistema é gerida por um processo e é dada prioridade às alterações mais urgentes.
- Processo de gestão de alterações
 - Analisar custos e benefícios das alterações propostas
 - Aprovar alterações que valem a pena
 - Gerir o rasto dos componentes do sistema que foram/são alterados



Gestão de versões

- Gestão de versões é o processo de manter o “rasto” das diferentes versões dos componentes de software, dos itens de configuração de e do sistema.
- Deve garantir que alterações feitas por diferentes programadores não interferem com o trabalho de outros programadores.
- Pode ser visto como um processo usado para gerir *codelines* e *baselines*



Codelines e baselines

- Um *codeline* é uma sequência de versões de código, onde as versões mais recentes são derivadas de versões mais antigas
- *Codelines* aplicam-se normalmente aos componentes do sistema
 - Dá origem a diferentes versões de cada componente
- Uma *baseline* é uma definição de um sistema específico
- A *baseline* especifica as versões dos componentes que são incluídos no sistema, junto com a especificação das bibliotecas usadas, ficheiros de configurações, etc.



Codelines e baselines

Codeline (A)



Codeline (B)



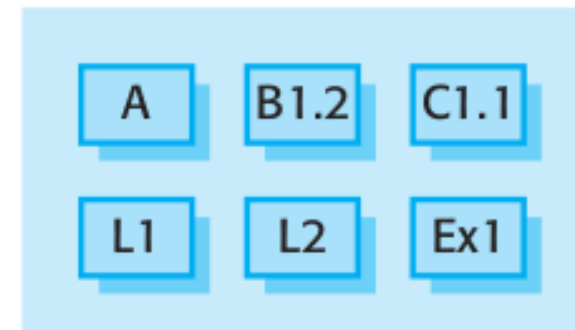
Codeline (C)



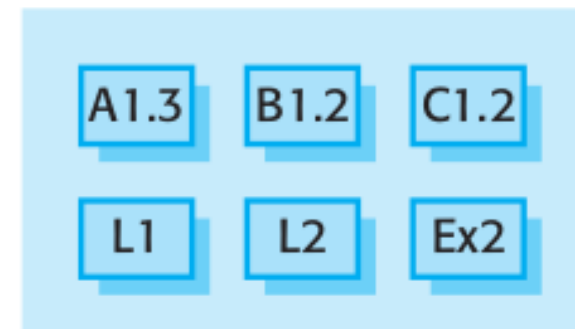
Libraries and External Components



Baseline - V1



Baseline - V2



Mainline



Baselines

- Especificação de *Baselines*
 - Recorrendo a linguagens/ferramentas específicas
 - Definir quais os componentes que fazem parte de uma versão do sistema
- Importância das *Baselines*
 - É necessário reconstruir uma versão específica do sistema
 - É necessário saber quais as versões exactas dos componente que compõem a versão do sistema



Sistemas de gestão de versões

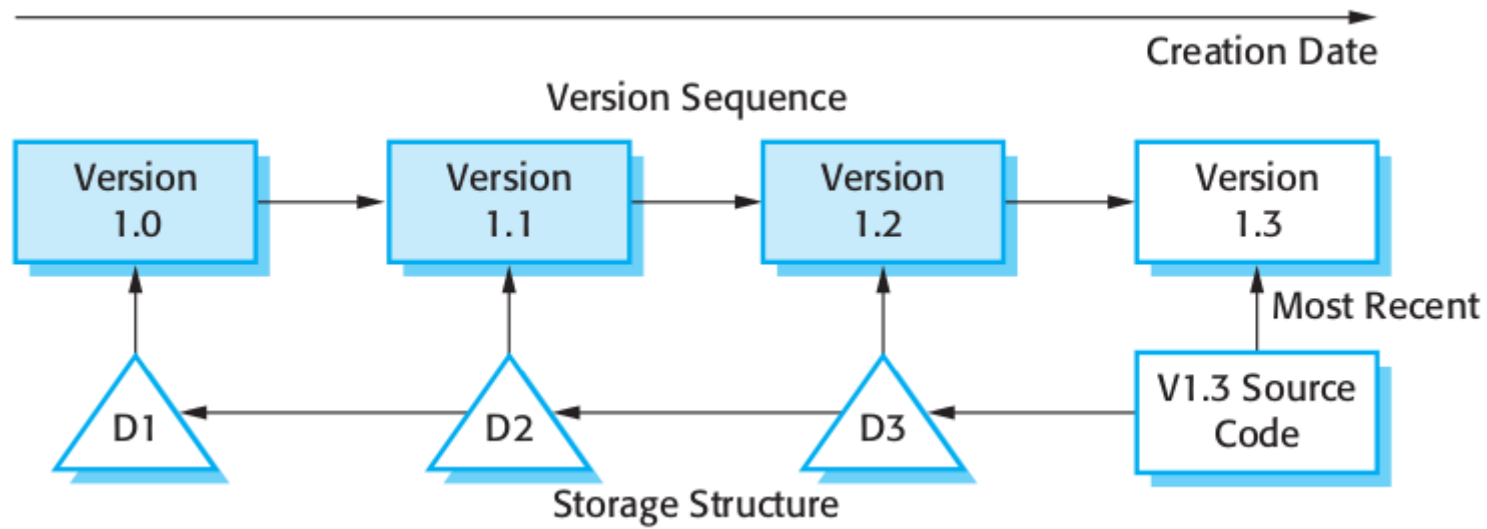
- Identificação de versões e *releases*
 - Cada versão tem um identificador único
- Gestão de armazenamento
 - Por forma a reduzir o espaço em armazenamento necessário para gerir todas as versões dos componentes, que podem variar muito pouco entre si, os sistemas de gestão de versões normalmente usam métodos que *agilizam* este armazenamento
- Registo de todas as alterações
 - Todas alterações feitas ao código devem ser registadas.



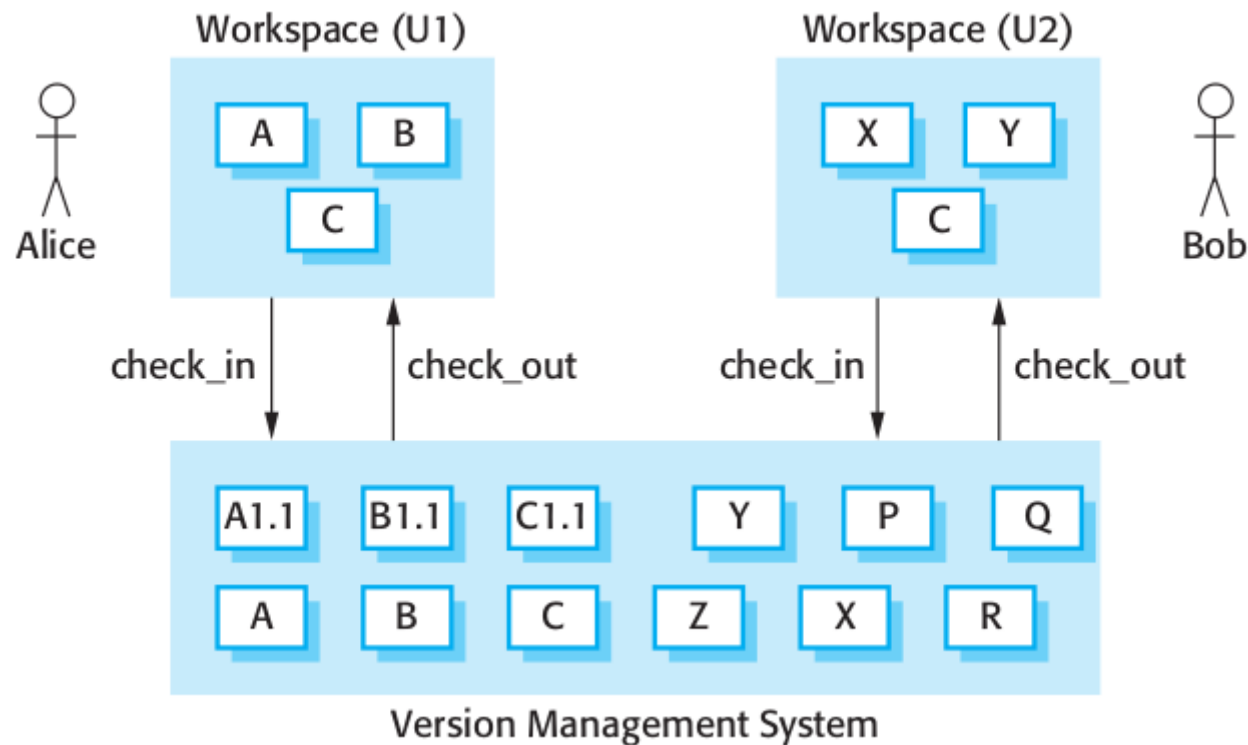
Sistemas de gestão de versões

- Desenvolvimento independente
 - Mantém registo dos componentes que foram alterados
 - Garante que alterações feitas no componente por outro programador não causa interferências
- Apoio ao projecto
 - Deve permitir a gestão do desenvolvimento de vários projecto que partilham componentes.

Sistemas de gestão de versões – Gestão do armazenamento



Sistemas de gestão de versões – *Check-in e check-out*



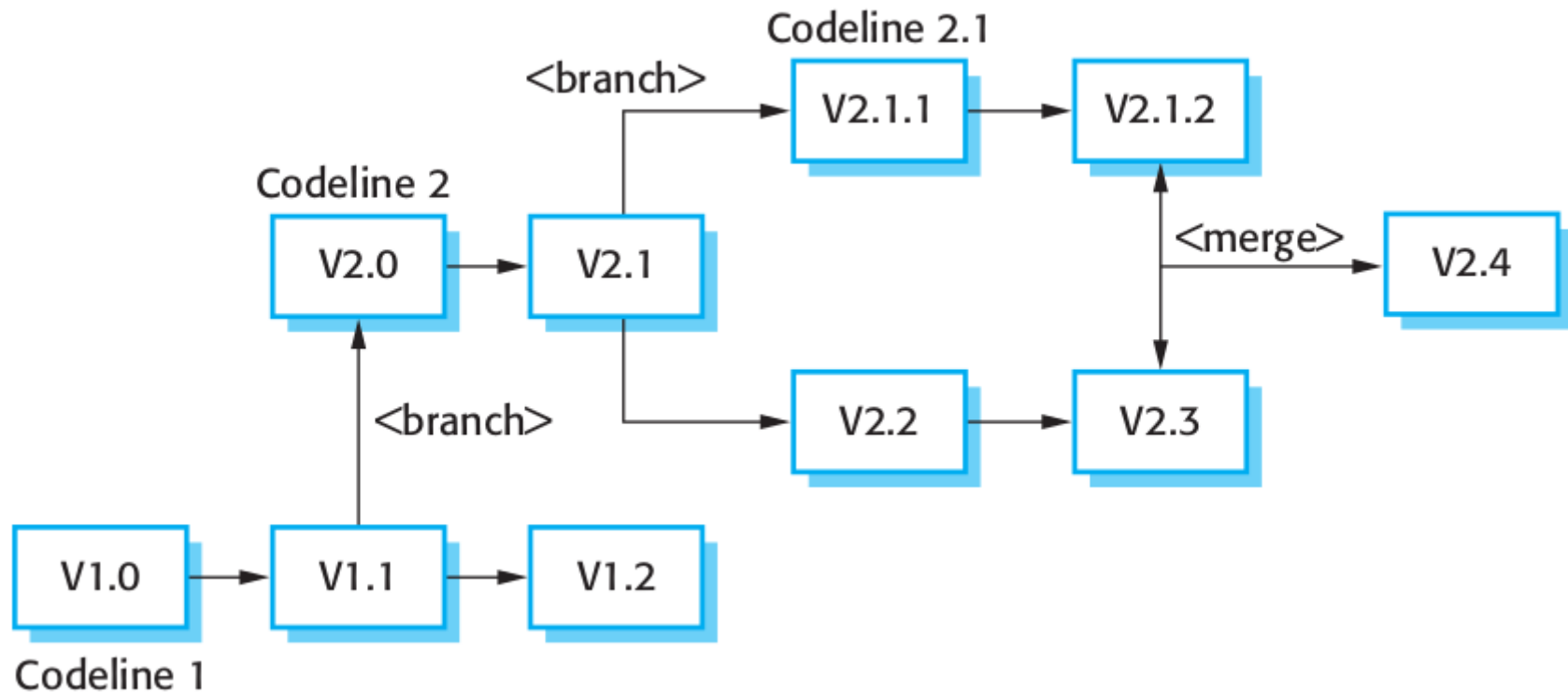


Sistemas de gestão de versões – Codeline branches

- Em vez de uma sequência linear de versões que reflectem alterações a um componente ao longo do tempo, podem existir várias sequências independentes.
 - Situação comum no desenvolvimento de um sistema, onde diferentes programadores trabalham de forma independente em diferentes versões do código, fazendo diferentes alterações.
- Em determinada etapa, será necessário juntar(merge) das *codeline branches* para criar uma nova versão do componente que inclua todas alterações feitas.
 - Se as alterações feitas envolverem diferentes partes do código, a junção(merge) pode ser feita de forma automática, combinando as diferenças(deltas) das diferentes versões.



Sistemas de gestão de versões – *Branching e merging*





Construção do sistema - System building

- Processo de criar uma versão completa e executável do sistema, compilando e *linkando* os vários componentes do sistema, bibliotecas externas, ficheiros de configuração, etc...
- Ferramentas de construção do sistema e ferramentas de gestão de versões devem comunicar entre si.
 - Fazer *check-out* de versões de componentes por fazer a construção do sistema
- Descrição da *baseline* deve ser feita na ferramenta de construção do sistema.

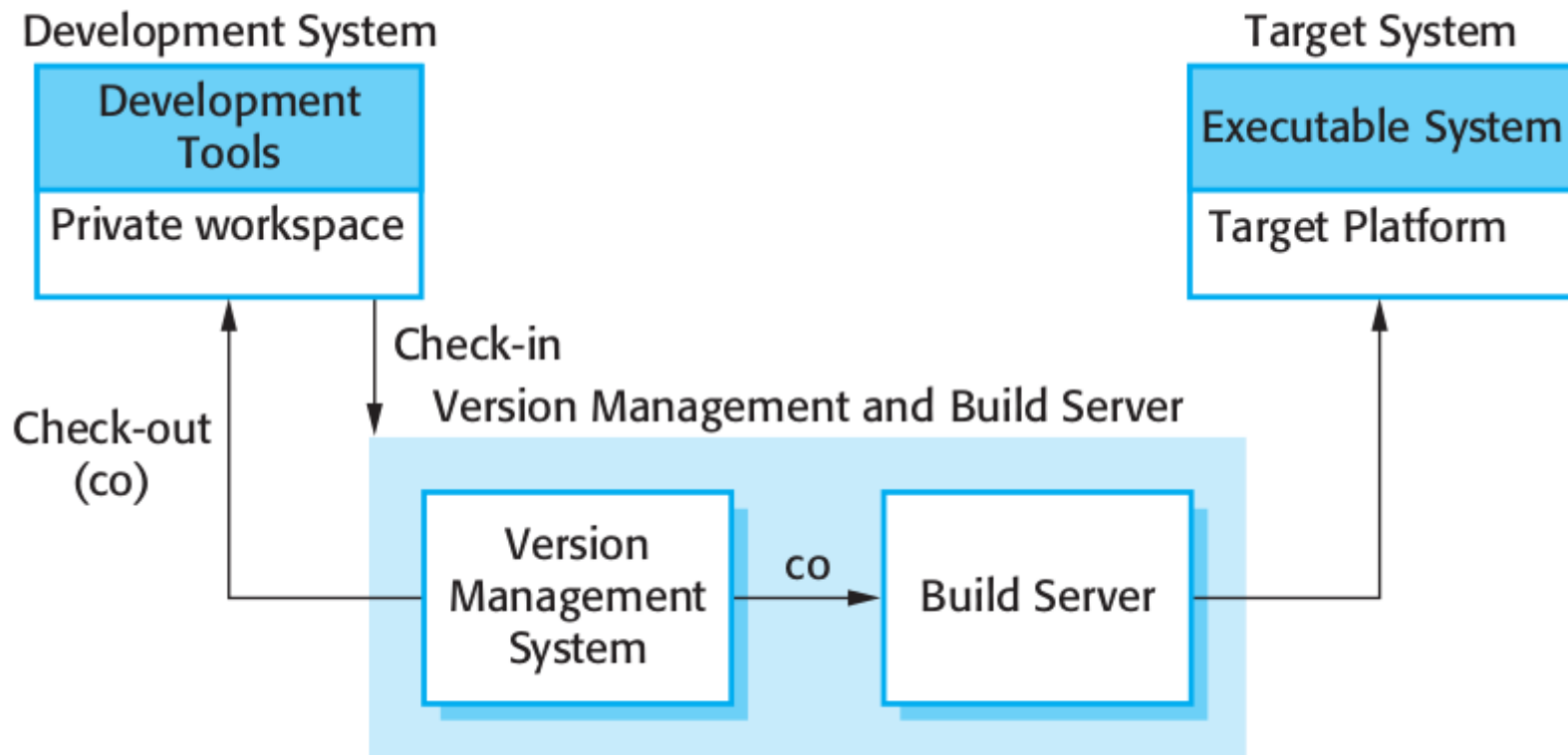


Plataformas de construção

- Plataformas de desenvolvimento
 - Incluem ferramentas de desenvolvimento
 - Compiladores, editores, etc
 - Programadores fazem checkout do código a partir do sistema de gestão de versões para um *workspace* privado antes de fazerem alterações ao sistema
- Build server
 - Usado para criar versões executáveis (releases) do sistema
 - Programadores fazem check-in do código para o sistema gestão de versões antes do *build* ser feito.
- Ambiente de produção
 - Plataforma onde o sistema é colocado em uso.

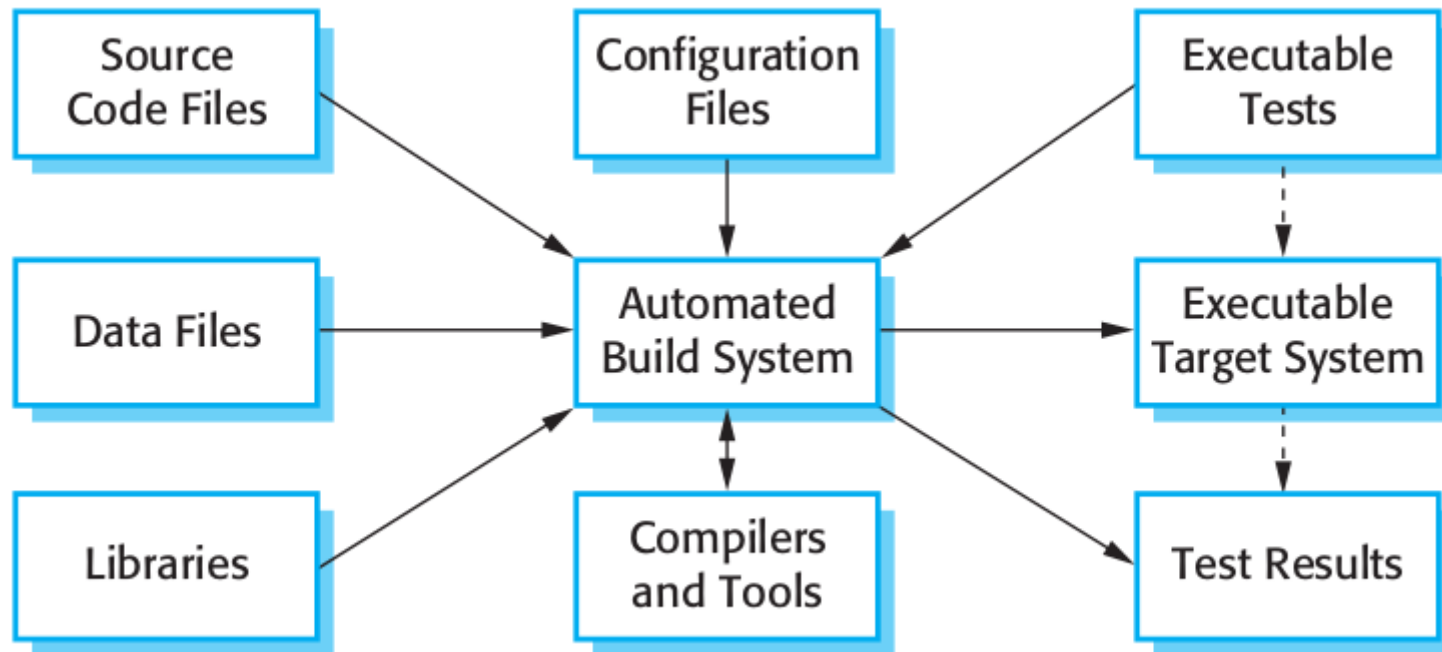


Plataformas de construção





Construção do sistema (system building)





Build system

- Funcionalidades
 - Script para gerar o *build*
 - Integração com o sistema de gestão de versões
 - Recompilação mínima
 - Criação de um sistema executável
 - Automação dos testes
 - Geração de documentação e relatórios



System build – Métodos ágeis

- Check-out da *mainline* a partir do sistema de gestão de versões para o ambiente privado de desenvolvimento do programador.
- Construir o sistema e correr todos os testes de forma automática para ter a certeza de que todos os testes passam com sucesso. Se não, a *build* tem problemas e deve informar-se o responsável pela ultima baseline (responsável por corrigir o problema).
- Fazer as alterações aos componentes do sistema.
- Construir o sistema no ambiente de desenvolvimento e correr todos os testes. Se os testes falharem, continuar o trabalho até os testes correm com sucesso.



System build – Métodos ágeis

- Depois de todos os testes passarem com sucesso, fazer *check in* no sistema de gestão de versões.
- Construir o sistema no *build server* e correr todos os testes.
 - Necessário se houve outros programadores que fizeram alterações desde o *check out* inicial
 - Se houver testes que falham, é necessário corrigi-los no ambiente de desenvolvimento pessoal do programador.
- Se os testes passarem no *build server*, fazer *commit* das alterações como uma nova *baseline* na *mainline* do sistema.