

INF016 – Arquitetura de Software

02 – Reorientação da Engenharia de Software

Sandro Santos Andrade
sandroandrade@ifba.edu.br

Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Eletro-Eletrônica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas



Arquiteturas em Contexto

- "o # o a arquitetura de software se relaciona # os # on # eitos de engenharia de software tradicionais # ente # a # i # ados "
- (ais # on # eitos de) # ser re*orientados + # ois a # arquitetura do software # essa a # u # ar # a # e # funda # enta #
- " onhe # i # entos funda # entais,
 - (oda a # i # ação te # u # a arquitetura
 - (oda a # i # ação te # ao # enos u # arquiteto
 - Arquitetura **não** é u # a fase do desen) o # i # ento

Arquiteturas em Contexto

- - uest . es %onsequentes,
 - / e onde surge a arquitetura de u# a a&\$i%ação '
 - " o# o u# a arquitetura de software &ode ser %ara%teri0ada '
 - - uais são as suas &ro&riedades '
 - 1 u# a arquitetura oa ou rui# '
 - Suas defi%i2n%ias &ode# ser fa%i\$# ente %orrigidas '
 - 3s arquitetos estão se# &re %ons%ientes das de%is . es funda# entais de &ro4eto que to# a# '
 - Essas de%is . es de &ro4eto &ode# ser arti%u\$adas %o# outras '

Arquiteturas em Contexto

- - uest . es %onsequentes,
 - 3s arquitetos %onsegue # # anter a integridade %on%eitua\$ do &ro4eto ao \$ongo do te # &o '
 - A\$ternati)as fora # %onsideradas nos di)ersos # o # entos de de%isão '
- A arquitetura do *software* não 5 &roducto de u # a fase es&e%6fi%a do &ro%esso+ rea\$i0ada a&7s a an8\$ise de requisitos e antes do &ro4eto deta\$hado
- A %riação e # anutenção da arquitetura estão &resentes e # todo o &ro%esso+ e # ora tenha # destaque es&e%ia\$ e # u # a fase &arti%u\$ar

Análise de Requisitos

- Considerações sobre a arquitetura e a implementação no início do projeto
- Noções de estrutura de projeto e solução são importantes durante a fase de análise de requisitos
- Visão tradicional, a análise e especificação de requisitos deve ser iniciada de qualquer consideração sobre qual projeto irá satisfazer os requisitos
 - A arte central deste artigo é a abordagem para a análise de requisitos que evita a atração negativa da orientação ; solução

Análise de Requisitos

- Esta ideia não é nova,

“Na análise nós começamos pelo que é requerido, o tomamos como certo e traçamos consequências até que alcancemos um ponto de onde podemos começar a síntese”

Aa&us+ s5%u\$o B /!"!

- A síntese aqui entende-se o projeto ou a elaboração da solução

Análise de Requisitos

- A ordem de George A. Balz e *How to solve it* * 1EFG,
 - A primeira regra é a
 - / e os elementos entre os dados e o desenhado! E a segunda é encontrar a solução
 - E há uma solução
 - E há uma solução
- A primeira ordem defende a elaboração e a reengenharia dos requisitos antes de resolver a solução

Análise de Requisitos

- É o uso desta abordagem, nas etapas de análise de requisitos, que não envolve a elaboração de requisitos; é a etapa de requisitos
- 3 fases nos requisitos: se qualquer alteração de requisitos; a alteração de requisitos, as alterações, as alterações rotacionais e os agitadores
- Na prática, entretanto, a análise de requisitos é realizada de modo iterativo e sucessivo,
 - Restrições de orçamento e recursos
 - Aumento de requisitos inferiores
 - Falta de confiança nos engenheiros responsáveis

Análise de Requisitos

- 3s # oti)os+ entretanto+ en)o\$)e # \$i# itaç . es hu # anas e # re\$ação a ra%ion6nio a strato+ e%ono # ia e e)o%ação
- Ana\$ogia %o # %onstrução de %asas,
 - Não ra%io%ina # os so re nossas ne%essidades inde&endente de %o # o e\$as serão satisfeitas
 - Aensa # os e # nJ # ero de %K # odos+ esti\$ das 4ane\$as+ fogão a g8s ou e\$5tri%o
 - Não &ensa # os e # ter # os de :u # a for # a de &ro)er a rigo a u # %\$i # a rigoroso<+ :u # a for # a de &ro)er i\$u # inação adequada< ou :u # a for # a de &re&arar %o # ida aque%ida<

Análise de Requisitos

- 3 # es # o a%onte%e %o # *software*,
 - Se # refer2n%ia a arquiteturas 48 eHistentes torna*se dif6%i\$a) a\$iar a)ia i\$idade+ %ronogra#a e %usto do &ro4eto
 - " onhe%er as interfa%es de usu8rio+ *hardware* e ti&os de ser)iços dis&on6)eis a4uda a %hegar e # requisitos aseados nu # a %o # &reensão ra0o8)e\$ da)ia i\$idade
 - As **falhas** i # &u\$iona # a engenharia e são a ase &ara ino)ação, o ser)ação L dete%ção das \$i # itaç . es
 - EHe # &\$o, %riação do *zipper* – su%essor de u # a \$onga sequ2n%ia de in)enç . es
 - : " o # o # uitos outros &rodutos+ o 0i&&er não surgiu direta # ente das fun%ionasidades # as de %orreç . es su%essivas de fa\$has< -Aetros?i 1EE2@

Análise de Requisitos

- 3 ser)aç . es funda# entais,
 - Aro4etos e arquiteturas 48 eHistentes define# u#)o%a u\$8rio &ara dis%utir as &ossi i\$idades
 - Nossa %o# &reensão so re o que fun%iona ho4e e %o# o e\$e fun%iona afeta nossos dese4os – fo%o na so\$ução
 - EH&eri2n%ias &r5)ias %o# siste# as nos a4uda a a) a\$iar a)ia i\$idade e definir %ustos e &ra0os
 - Requisitos M arti%u\$ação de # e\$horias ne%ess8rias ; arquitetura)igente
 - Isso não signifi%a \$i# itar ino)ação+ as # 8quinas de \$a) ar fora# &rogressi)a# ente a&erfeioadas

Análise de Requisitos

- 3 ser)aç . es funda# entais,
 - Não se \$i# itar+ entretanto+ aos &ro4etos atuais! / iferentes # e%anis# os de)e# ser usados &ara su ir e# u#a %asa+ arranha*%5u ou at5 a \$ua
 - - uando não eHiste# ante%essores f6si%os ana\$ogias ou ante%essores %on%eituais &ode# a4udar
 - / esen)o\$)i# ento *greenfield* ta# 5# uti\$ioa ante%essores &ara enquadrar os requisitos e so\$uç . es in5ditos
 - Ne# todas as arquiteturas+ entretanto+ são oas fontes de ins&iração

Projeto

- Fase onde se dá maior atenção à definição das necessidades de arquitetura
- 30% do tempo da arquitetura, entretanto, não é o uso desta fase
- A arquitetura é um aspecto de todas as outras atividades de desenvolvimento
- As decisões arquiteturais refletem os requisitos do sistema, requerendo um compromisso de prioridades de projeto

Projeto

- I o de \$o tradi%iona\$ Ne # %as%ata0,
 - 3 4eti)o, definir u# &ro4eto que &ossa ser re&assado &ara os &rogra#adores
 - Se a\$gu# requisito 5 %onsiderado in)i8)e\$+ retorna*se ; fase de an8\$ise de requisitos Nse# entretanto reto# ar as&e%tos da so\$ução0
 - Se+ na i# &\$e# entaçãõ+ a\$gu# a &arte do &ro4eto 5 %onsiderada in)i8)e\$+ retorna*se ; fase de &ro4eto

Projeto

- **Modelos de entrada e saídas arquiteturas,**
 - Redução ou eliminação das fronteiras entre as fases geradas entre artificiais e produtivas
 - Análise de requisitos relacionada aos aspectos de arquitetura e projeto
 - Análise de projeto e implementação a partir de uma forma mais integrada e enriquecida
- **3 arquitetura usada para a análise de questões,**
 - Interesses dos *stakeholders*, estilo e estrutura utilizados, tipos de componentes, estrutura físicas de classes e objetos, aspectos de distribuição, desempenho, segurança, etc

Projeto

- (5) fases de projeto de sistemas,
 - Projeto orientado a objetos,
 - Identificação de objetos que em um estado e funções que acessa e manipula este estado
 - Não é uma abordagem orientada a objetos e todas as situações! Não é ineficiente, entretanto
 - Práticas,
 - Não é uma abordagem orientada a objetos! Não a abordagem de identificação, segurança, confiança, etc
 - Não possui mecanismos para transferir e reutilizar as arquiteturas, portanto de domínio e soluções presentes em arquiteturas anteriores
 - Exige que todos os objetos e entidades sejam objetos! Não há suporte explícito para algo que não seja um objeto

Projeto

- (5) fases de projeto de sistemas,
 - Projeto orientado a objetos,
 - Principais,
 - / is&oni is&oni a so# ente u# ti&o de en%a&su\$a# ento No 4eto0+ u# a noção de interfa%e+ u# Jni%o ti&o de %one%tor eH&\$6%ito Nprocedure call0! Não su&orta a noção de *required interfaces*
 - Forte# ente sigada a interesses e de%is . es das \$inguagens de &rogra# ação+ que &ode# %o# eçar a ditar quais de%is . es são i#&ortantes
 - Assu# e u# es&aço de endereça# ento %o# &arti\$hado e su&orte adequado ao geren%ia# ento de *heap* e *stack*
 - Assu# e a eHist2n%ia de u# a Jni%a *thread* de %ontro\$e
 - As&etos de %on%orr2n%ia+ distri uição e des%entra\$ão não são %onsiderados
 - A Q I P ajudou a dis%utir u# &ro4eto orientado a o 4etos se# de&ender da \$ingua#e# de &rogra# ação

Projeto

- (5) fases de projeto de sistemas,
 - *Domain-Specific Software Architectures* / *SSAs*
 - Adotada quando existem e arquiteturas anteriores influenciam e orientam os projetos
 - Carente a uma boa abordagem e melhor solução para o domínio e questão
 - Novas arquiteturas serão variações das anteriores
 - A re-estruturação para formas originais e variáveis
 - Reutilização de partes da arquitetura e da implementação
 - High-level suporte, arquiteturas anteriores devem ser adaptadas e refinadas para reuso e pontos de variação devem ser identificados e isolados, interfaces nos pontos de variação devem ser explicitadas e

Implementação

- 3 4eti)o, %riar u# %7digo*fonte que se4a fie\$; arquitetura e que i# &\$e # enta de for # a %o # &\$eta os requisitos
- A a ordage# %entrada e# arquiteturas d8 2nfase a a\$gu# as a ordagens ; i# &\$e # entação,
 - A i# &\$e # entação &ode estender ou # odifi%ar a arquitetura
 - A arquitetura s7 estar8 %o # &\$eta a&7s a i# &\$e # entação
 - / e)e*se # anter as de%is . es que %onstitue# a arquitetura %onsistentes %o # o %7digo*fonte &rodu0ido
 - Esti# u\$a a uti\$i0ação de t5%ni%as generati)as e forte # ente aseadas e# reuti\$i0ação NeH, uso de *frameworks*0

Implementação

- Implementação física,
 - (Todos os elementos estruturais da arquitetura estão implementados no código-fonte)
 - 3 código-fonte não de) e utilizar elementos opcionais que não estão presentes na arquitetura
 - 3 código-fonte não de) e manter apenas os elementos da arquitetura que não estão desativados na arquitetura

Implementação

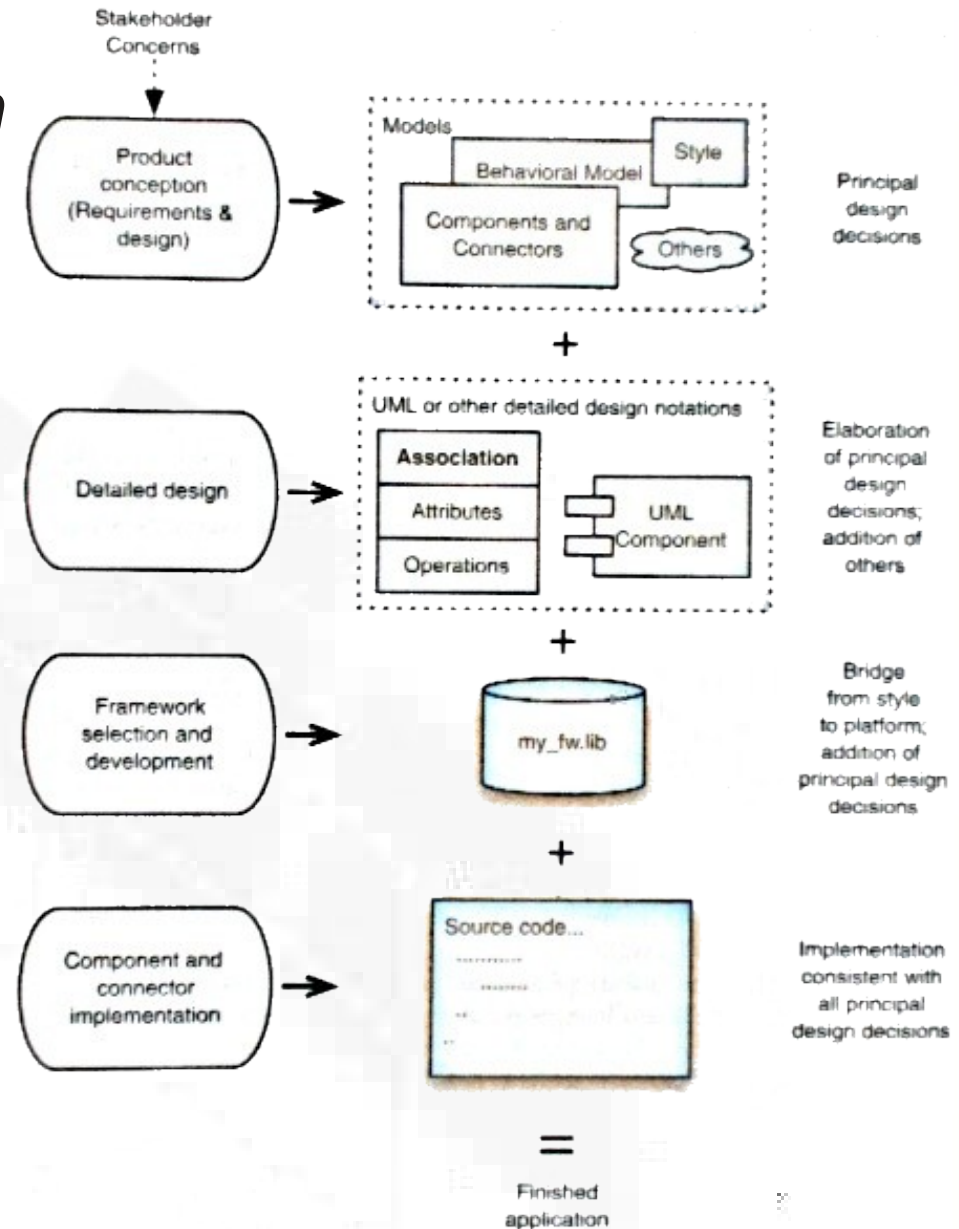
- Na &r8ti%a essa definição não 5 tota\$# ente adequada,
 - Qso de i \$iote%as redu0e# o %usto e au#enta# a qua\$idade+ &or5# %ont2# func. es e interfa%es que não estão &resentes na arquitetura
 - Se u#a i \$iote%a arata e de qua\$idade i# &\$e#enta ER S da fun%iona\$idade dese4ada e se as %onsequ2n%ias da aus2n%ia dos outros 2S fore# a%eit8)eis,
 - / e%ide*se &e\$ uso da i \$iote%a
 - Rea\$i0a*se a re)isãõ das es&e%ifi%aç. es e da arquitetura
 - 3 &onto %r6ti%o 5 se#&re # anter a arquitetura e a i# &\$e#entação e# estados %onsistentes

Implementação

- Estratégias de implementação e prioridade,
 - 10 (5%ni%as generati) as, a i# &\$e # entação 5 gerada auto # ati%a # ente e &ossui a\$ta qua\$idade
 - Cera\$ # ente a&\$i%ada e # do # 6nios # uito es&e%6fi%os
 - 20 (5%ni%as aseadas e # reuso, de # anda # # enor te # &o e &rodu0e # %7digo de # aior qua\$idade do que %onstruir o siste # a do 0ero
 - *Architecture-Implementation Framework* M &onte entre u # esti\$o arquitetura\$ es&e%6fi%o e u # %on4unto de te%no\$ogias de i# &\$e # entação! (ra0 garantias
 - So\$uç . es de *middleware*+ " 3 (S+ *open-source*
 - T0 / esen)o\$)i# ento # anua\$ %o # &\$eto, %ustos e &ra0os # aiores! I aior ne%essidade de garantia da qua\$idade

Implementação

■ *Architecture-Implementation Framework,*



Implementação

- Se a implementação difere da arquitetura projetada, esta arquitetura não caracteriza a ação
- 3 sistemas ou a arquitetura, ou o sistema, e o contraste; que do sistema
- Faz e reconhece esta diferença,
 - Rou a habilidade de raciocinar no futuro, sobre a arquitetura projetada da ação
 - Engana os *stakeholders* e reação ao que eles acreditam que é e o que eles realmente é
 - (orna qualquer estratégia de desenhamento) ou implementação baseada na arquitetura e relacionada ao fracasso

Análise e Teste

- Atividades realizadas para garantir a qualidade de um artefato
- Na abordagem tradicional o código-fonte é analisado e testado para garantir a correção funcional e eventualmente desempenho
- Entretanto, a análise de um artefato e a determinação da sua qualidade pode ser realizada assim que o artefato existir, seja ele o que for
- Porque só o código-fonte é testado
- Porque o teste é realizado somente em relação aos requisitos funcionais da aplicação

Análise e Teste

- Resposta, devido ; ausência de qualquer representação suficiente e rigorosa da ação que não seja o código-fonte
- Arquiteturas e iterativas a análise antecipada e a análise melhorada do código-fonte
- A resequência para a análise de propriedades não-funcionais
- - quais os benefícios que as arquiteturas de *software* trazem ; fase de análise e teste

Análise e Teste

10.3 #ode\$ arquitetura\$ &ode ser a) a\$iado e # re\$ação ; sua %onsist2n%ia interna e %orretude,

- 9erifi%aç . es sint8ti%as do # ode\$ &ode # identifi%ar+ &or eHe # &\$o+ %oneH . es entre %o # &onentes não %o # &at6) eis *interface mismatch*0+ es&e%ifi%ação in%o # &\$eta de &ro&riedades e &adr . es de %o # uni%ação indese4ados
- An8\$ise de f\$uHo de dados &ode ser a&\$i%ada &ara deter # inar in%o # &ati i\$idades de definiçãoUuso e &ara dete%tar fa\$has de segurança
- (5%ni%as de *model-checking* &ode # ana\$isar &ro \$e # as de *deadlock*
- (5%ni%as de si # u\$ação &ode # rea\$i0ar for # as si # &\$es de an8\$ise dinV # i%a

Análise e Teste

- 20 3 # ode\$ arquitetura\$ &ode ser a) a\$ iado e # re\$ ação ; sua %onsist2n%ia %o # os requisitos,
- Inde&endente do &ro%esso uti\$ i0ado o # ode\$ arquitetura\$ de) e ser %onsistente %o # os requisitos
 - Esta)erifi%ação ta\$) e0 &re%ise ser feita de for # a # anua\$+ %aso os requisitos este4a # des%ritos e # \$ingua\$e # natura\$
 - A)erifi%ação 5 essen%ia\$

Análise e Teste

T0 3 #ode\$ arquitetura\$ &ode ser uti\$i0ado &ara deter#inar e su&ortar estrat5gias de an8\$ise e teste a&\$i%adas ao %7digo*fonte,

- A arquitetura &ro)2 o &ro4eto do %7digo*fonte+ %onsist2n%ia entre e\$es 5 essen%ia\$
- A arquitetura ser)e %o#o u#a fonte de infor#ação &ara go)ernar testes+ aseados na es&e%ifi%ação+ que atua# e# todos os n6)eis, unidade+ su *siste#a e siste#a
 - 3 arquiteto &ode &riori0ar an8\$ises e testes %o# ase na arquitetura+ fo%ando os %o#&onentes e #ontagens #ais %r6ti%os
 - EH, " o#&onentes %o#uns e# todos os #e# ros de u#a fa#6\$ia de &rodutos

Análise e Teste

T0 3 #ode\$ arquitetura\$ &ode ser uti\$i0ado &ara deter#inar e su&ortar estrat5gias de an8\$ise e teste a&\$i%adas ao %7digo*fonte,

- A arquitetura ser)e %o#o u#a fonte de infor#ação &ara go)ernar testes+ aseados na es&e%ifi%ação+ que atua# e# todos os n6)eis, unidade+ su *siste#a e siste#a
 - A arquitetura dis&oni i\$i0a u# #eio &ara re&assar+ &ara no)os &ro4etos+ resu\$tados &r5)ios de an8\$ise
 - EH, a eHtensão do teste de unidade de u# %o#&onente 5 redu0ido se e\$e est8 sendo a&\$i%ado e# u# #es#o %onteHto e %ondiç .es de uso
 - A arquitetura a4uda a des)iar a atenção do ana\$ista &ara os %one%tores e# u#a i#&\$e# entação do siste#a

Análise e Teste

B0 3 # ode\$ arquitetura\$ &ode ser %o# &arado %o# u# # ode\$ deri)ado a &artir do %7digo*fonte da a&\$i%ação,

- 1 u# a for # a de %he%ar a sua so\$ução
- Se4a P u# &rogra # a deri)ado da arquitetura A
- Q # gru&o diferente de engenheiros+ se # a%esso a A + desen)os)e u# # ode\$ arquitetura\$ A' a &artir da an8\$ise de P
- Se tudo esti)er %orreto A ser8 %onsistente %o# A'
- " aso %ontr8rio+ ou P não i# &\$e # enta A fie\$ # ente ou A' não ref\$ete fie\$ # ente a arquitetura de P
- E # qua\$quer %aso 5 ne%ess8rio u# a)erifi%ação

Evolução e Manutenção

- Evolução ou Manutenção de *Software* refere-se a todo tipo de atividade realizada a partir do momento de liberação da aplicação
- A abordagem tradicional e a evolução *ad-hoc* geram, em geral, uma retroalimentação; mudança
- Há risco de degradação da qualidade da aplicação,
 - devido a mudanças realizadas e quaisquer alterações ou quaisquer meio que se faça o sistema e os usuários
 - "o tempo de reação a mudanças sucessivas se torna entre si diferente de isto que de fato é o que se tem entre mudanças sucessivas anteriores"; torna

Evolução e Manutenção

- A a ordage # %entrada e # arquitetura ofere%e u # a ase s7\$ida &ara u # a e) o\$ução efi%a0,
 - Fo%o sustentado e # u # # ode\$o arquitetura\$ eH&\$6%ito+ rea\$ e # odifi%8) e\$
- Fases do &ro%esso de e) o\$ução,
 - 10 I oti)ação
 - 20 A) a\$iação
 - T0 Es%o\$ha e &ro4eto da a ordage #
 - B0 EHe%ução+ in%\$uindo a &re&aração &ara a &r7Hi # a rodada de ada&tação

Evolução e Manutenção

10) Iotação,

- / entre as di)ersas # oti)aç . es &ara e) o\$uçãõ+ desta%a*se a %riação de no)as)ers . es de u# &roduto
- >ustifi%a o estudo de fa # 6\$ias de &roduto

20) A) a\$iação,

- A # udança 5 eHa # inada &ara deter # inar sua)ia i\$idade! " aso se4a)i8) e\$+ %o # o e\$a ser8 a\$%ançada '
- Requer %onhe%i # ento a&rofundado so re o &roduto e # questão
- Se u# # ode\$o arquitetura\$ fie\$; i# &\$e # entação est8 dis&on6) e\$ a %o # &reensão e an8\$ise da # udança o%orre # de for # a efi%a0

Evolução e Manutenção

T0 Especificação e projeto da arquitetura,

- A arquitetura deve satisfazer todos os requisitos que motivaram a mudança
- 1 realizada durante a especificação

B0 Evolução,

- 3 requisitos artefatos a serem desenvolvidos o modelo da arquitetura
- 7 então mudanças no código-fonte são realizadas
- A consistência deve ser preservada! Ferramentas ajudam nesta tarefa
- A manutenção não está concluída até que os modelos estejam consistentes

Processos de Desenvolvimento

- A arquitetura &er # eia todas as ati)idades de desen) o\$) i # ento e # anutenção do *software*
- 3 %on4unto de de%is . es arquiteturais 5 %riado e # %onsonVn%ia %o # os requisitos e %ontinua se eH&andindo at5 a # anutenção
- A nature0a %entra\$ da arquitetura 5 o s%ure%ida nas %ara%teri0aç . es tradi%ionais das ati)idades de desen) o\$) i # ento de *software*,
 - As ati)idades do &ro%esso são o &onto %entra\$+ não en%ontra*se a arquitetura e # nenhu# \$ugar
 - EHige \$i # ites r6gidos entre os ti&os de ati)idades

Processos de Desenvolvimento

- Afirma-se que o processo de desenvolvimento não quer dizer que existe uma forma correta de realizar o desenvolvimento,
 - Estratégias e artefatos de desenvolvimento se beneficiam dos pontos fortes e referências da organização
 - Diferentes ambientes de modificação irão demandar maior atenção a certas atividades
- Quando for analisado o comportamento e o desempenho de diferentes estratégias de desenvolvimento, é necessário considerar as atividades das arquiteturas

Processos de Desenvolvimento

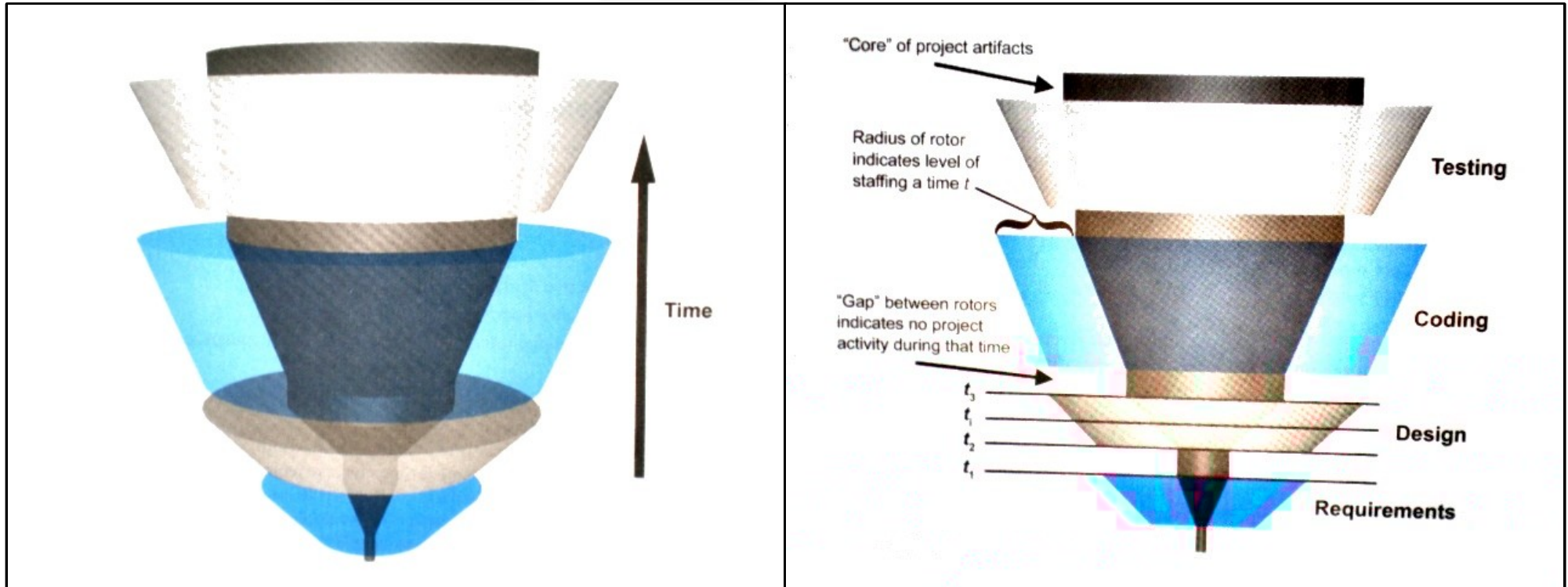
- 9 isua\$ i0ação e # (ur ina,
 - I 5todo &ara i\$ustrar u# %on4unto integrado de ati)idades de desen)o\$)i# ento no qua\$ 5 e)iden%iado o &a&e\$ %entra\$ da arquitetura do *software*
- Pe)a e# %onsideração diferentes as&e%tos do desen)o\$)i# ento,
 - (e# &o
 - (i&os de ati)idades ati)as e# u# deter#inado te# &o
 - Esforço NeH, horas tra a\$hadas0 e# u# deter#inado te# &o
 - Estado do &roduto NeH, %onteJdo gera\$ do &ro4eto ou %onhe%i# ento a%u# u\$ado0

Processos de Desenvolvimento

- 9 isua\$ioação e# (ur ina,
 - 1 definida es&a%ia\$# ente &or u# %on4unto de an5is %o# \$argura e es&essura)ari8)eis+ dis&ostos ao \$ongo de u# nJ%\$eo
 - 3 eiHo do nJ%\$eo 5 o te# &o
 - 3 nJ%\$eo re&resenta o &roduto sendo desen)o\$)ido
 - 3s an5is re&resenta# fases de\$# itadas &e\$ o te# &o
 - A es&essura do ane\$ denota a duração das N&ossi)e\$# ente %on%orrentes0 ati)idades do ane\$
 - 3)o\$u# e do ane\$ re&resenta o in)esti# ento rea\$ioado
 - Se%ç . es trans)ersais de u# # es# o ane\$ &ode# ser diferentes e inter)a\$os entre an5is &ode# eHistir

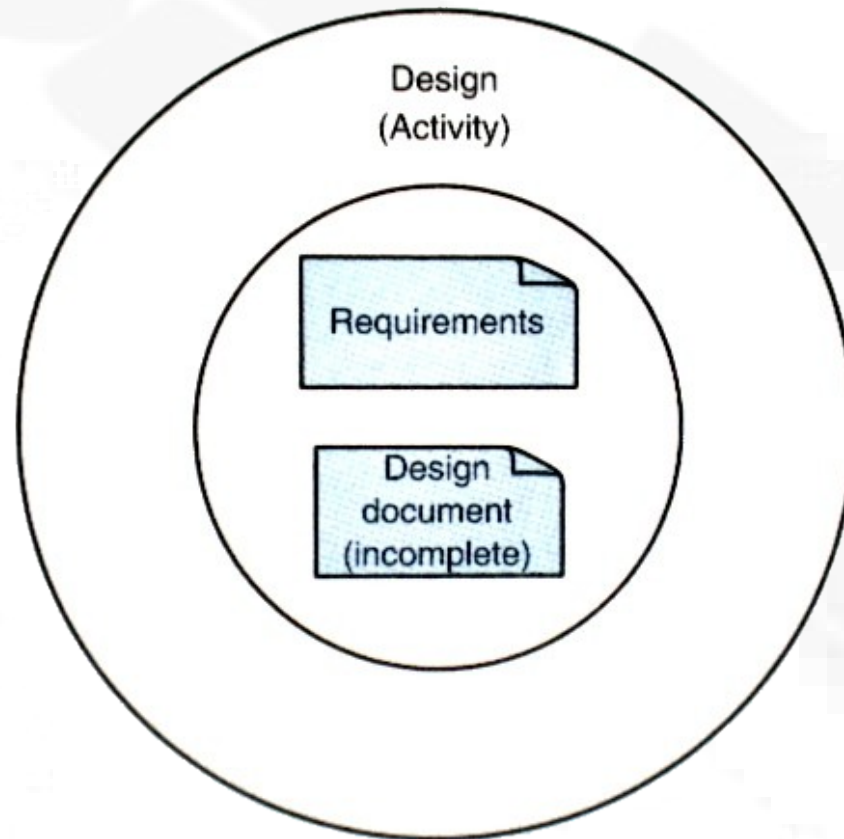
Processos de Desenvolvimento

- 9isua\$ioação e# (ur ina N#ode\$o e# %as%ata0,



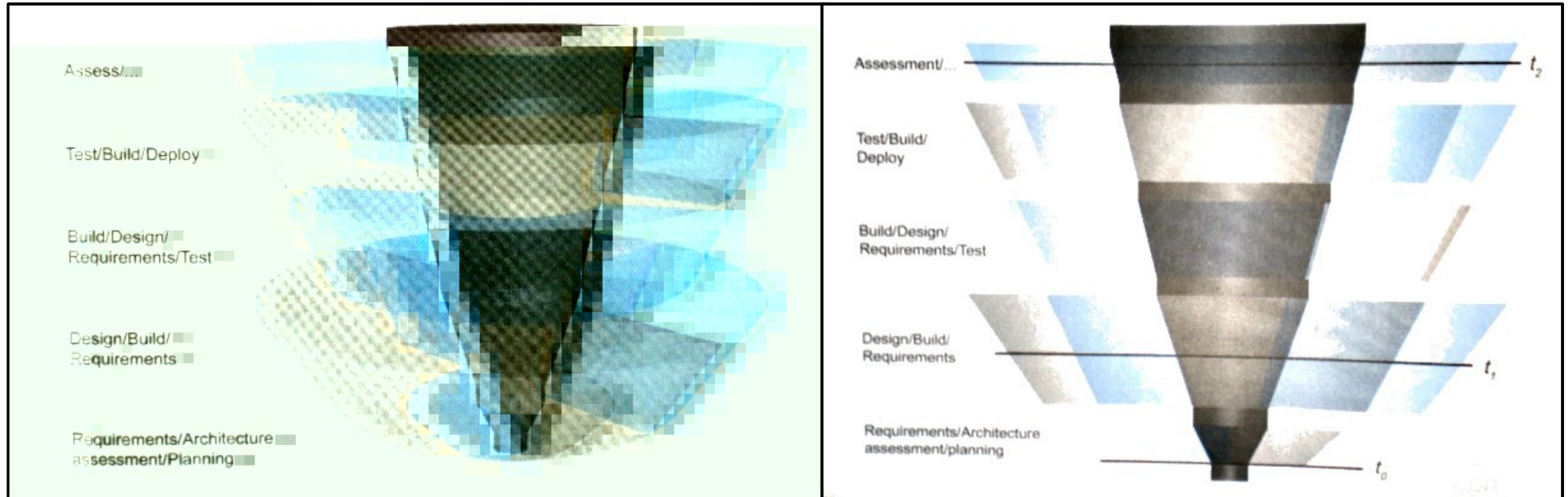
Processos de Desenvolvimento

- 9 isua\$ i0ação e# (ur ina N# ode\$ e# %as%ata0,
 - Se%ção trans)ersa\$ no te# &o t_1



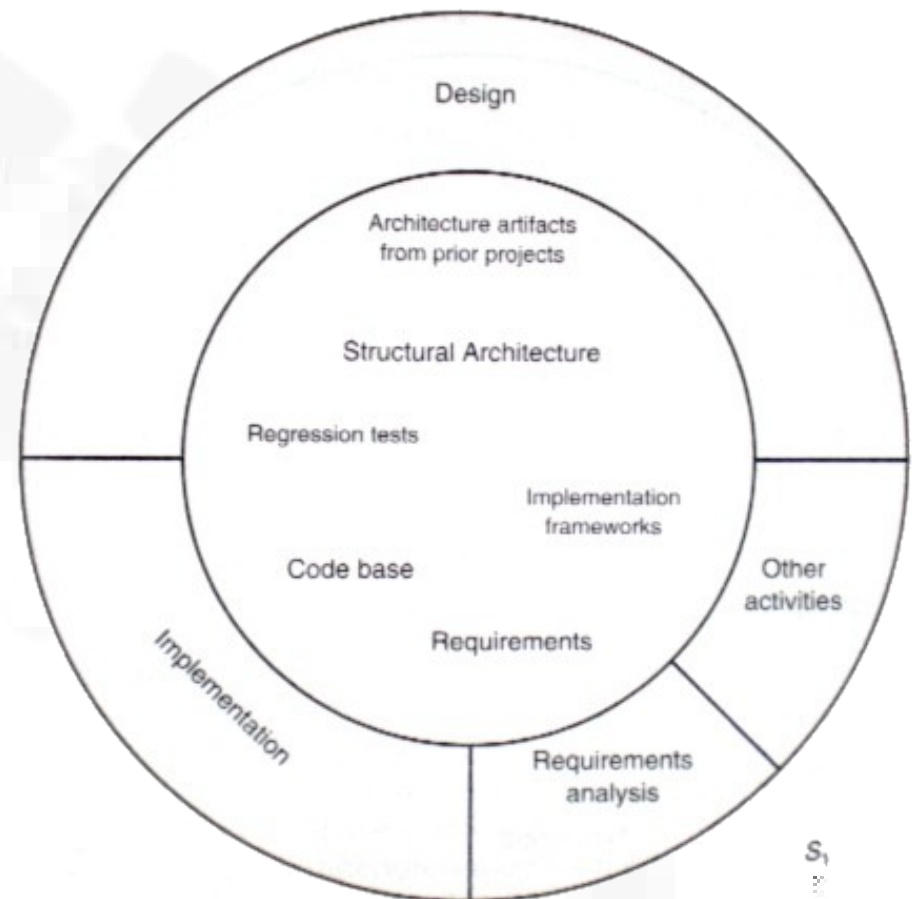
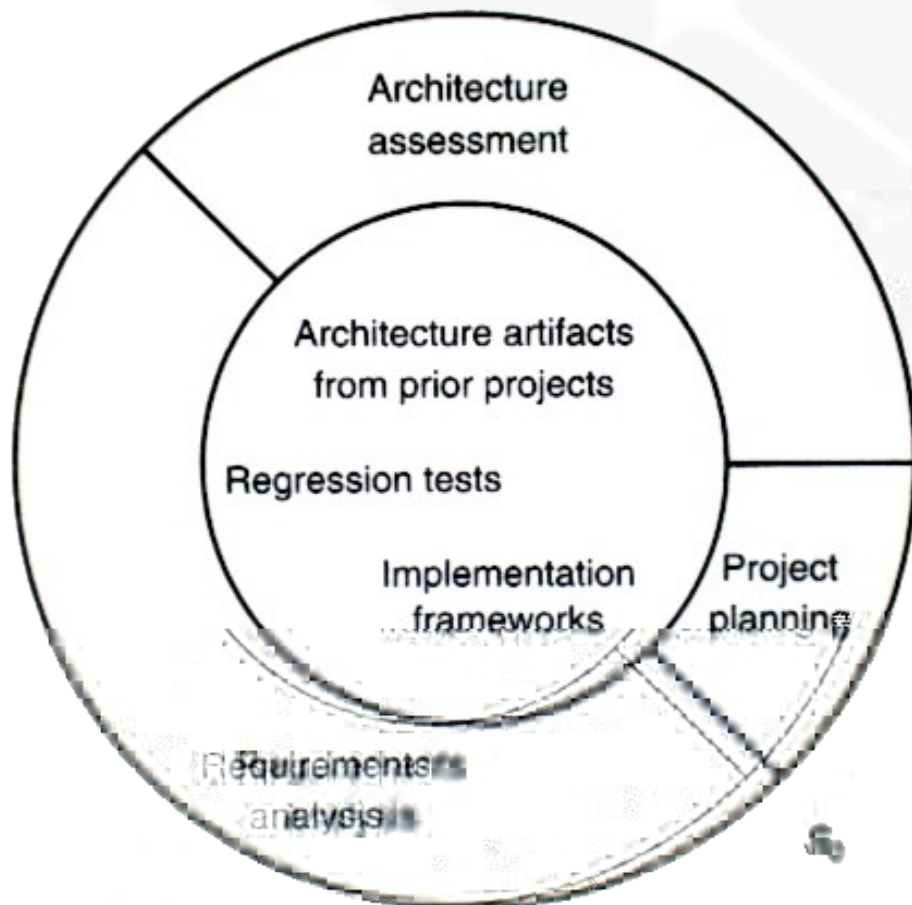
Processos de Desenvolvimento

- 9 isua\$ i0açaõ e# (ur ina Nati) idades %on%orrentes0,



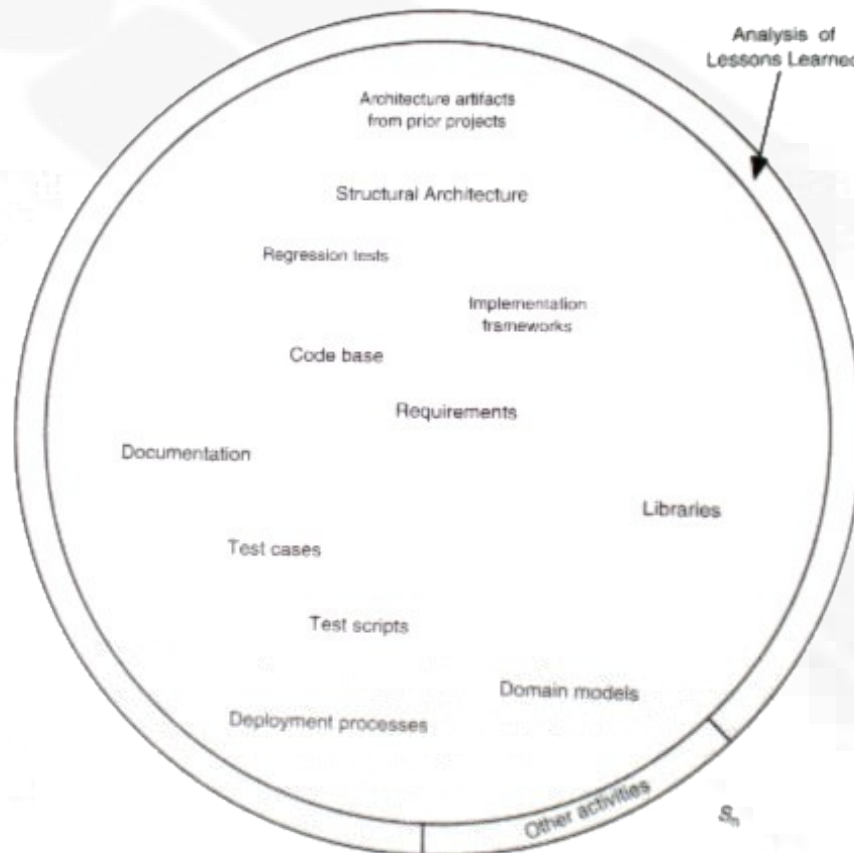
Processos de Desenvolvimento

- 9 isua\$ i0ação e # (ur ina Nati) idades %on%orrentes0,
 - Se%ç . es trans) ersais nos te # &os t_0 e t_1



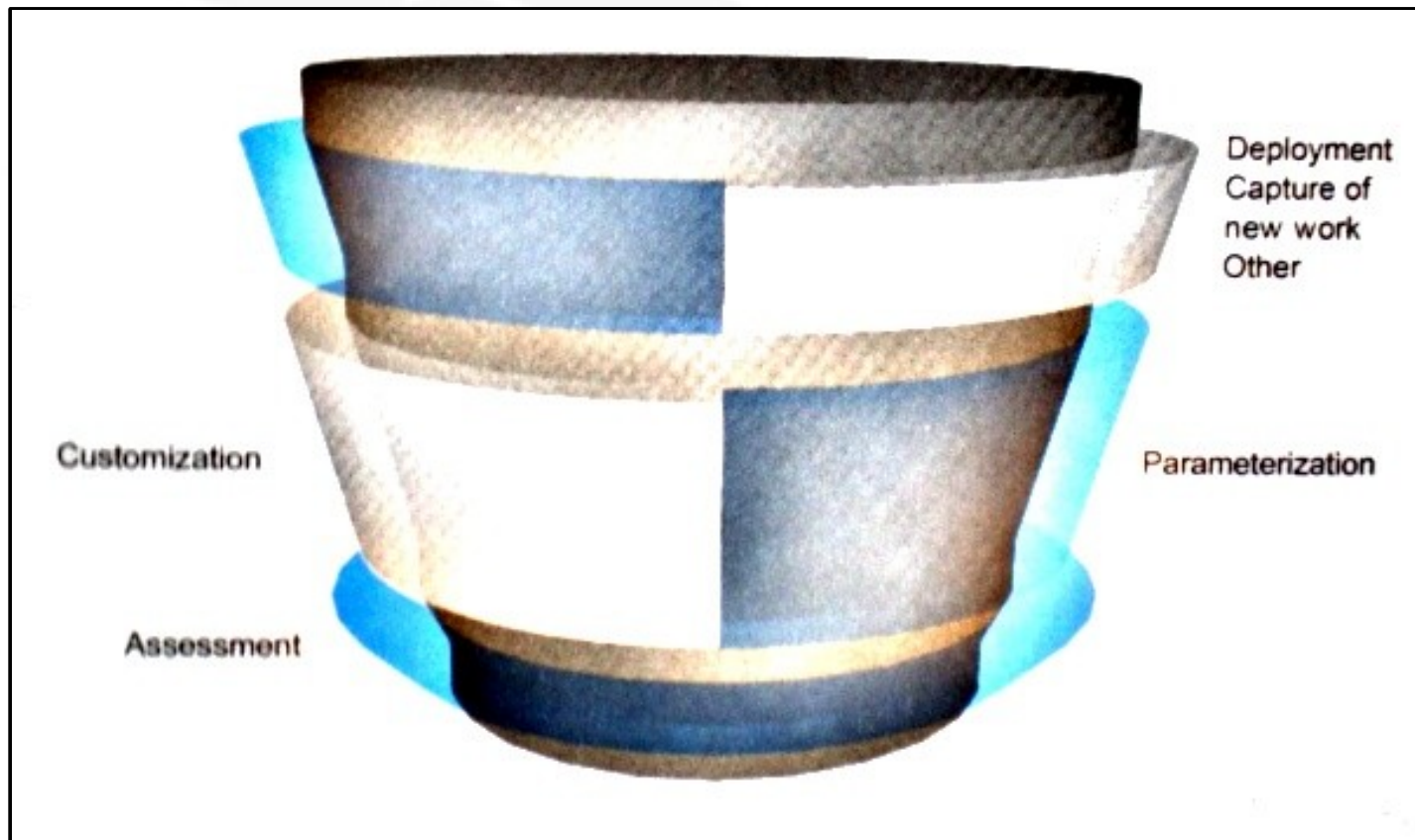
Processos de Desenvolvimento

- 9 isua\$ i0ação e # (ur ina Nati) idades %on%orrentes0,
 - Se%ção trans)ersa\$ &r7Hi# o ao fi# do &ro4eto



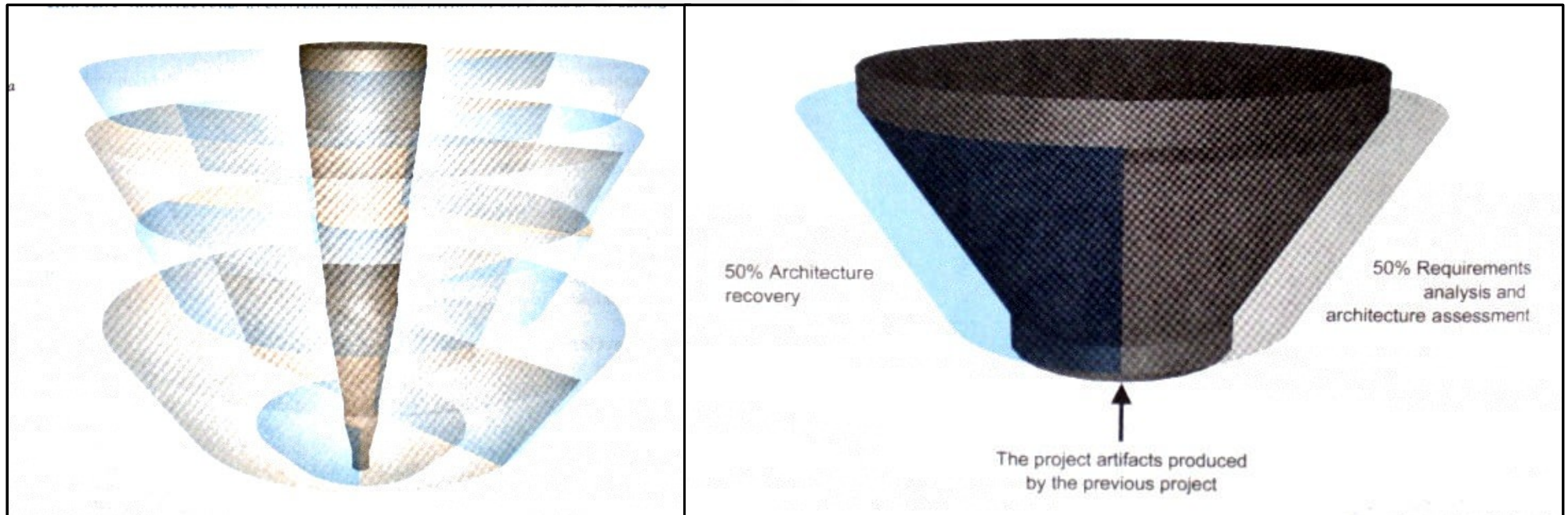
Processos de Desenvolvimento

- EHe # &\$o de des%rição de &ro%esso,
 - *Domain-Specific Software Architecture* N / SSA0



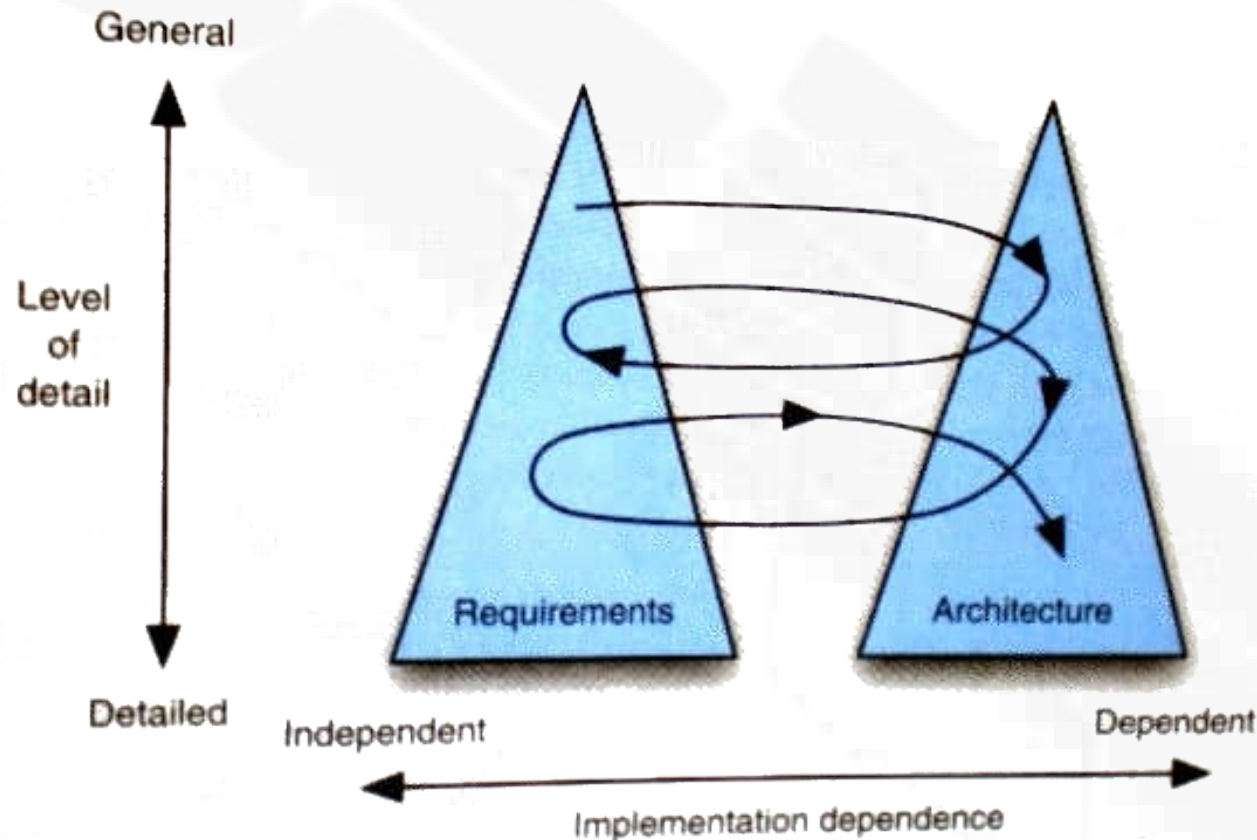
Processos de Desenvolvimento

- EHe # &\$o de des%rição de &ro%esso,
 - / esen)o\$)i# ento 8gi\$



Processos de Desenvolvimento

- Outros modelos de processo,
 - *Twin Peaks* (Nusei et al. 2001)



INF016 – Arquitetura de Software

02 – Reorientação da Engenharia de Software

Sandro Santos Andrade
sandroandrade@ifba.edu.br

Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Eletro-Eletrônica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas

