The integrated business change life cycle

ProcessForum

Changing the World - One Process at a Time

Etienne Venter, Santam

8th June 2011

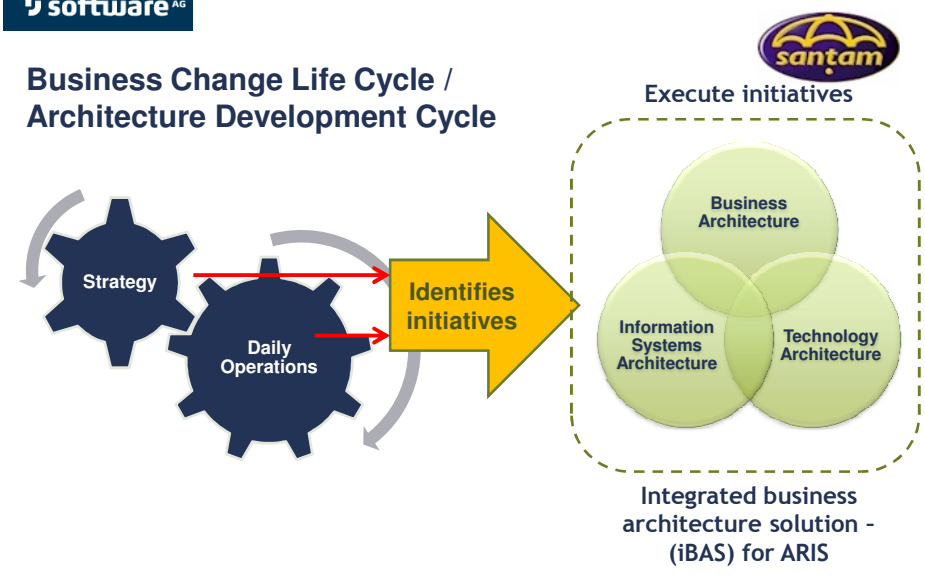
CONTENTS

Business Change Life Cycle / Architecture Development Cycle

The purpose of enterprise architecture is to optimize across the enterprise the often fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy. TOGAF

"An effective enterprise architecture is critical to business survival and success and is the indispensable means to achieving competitive advantage through IT." TOGAF

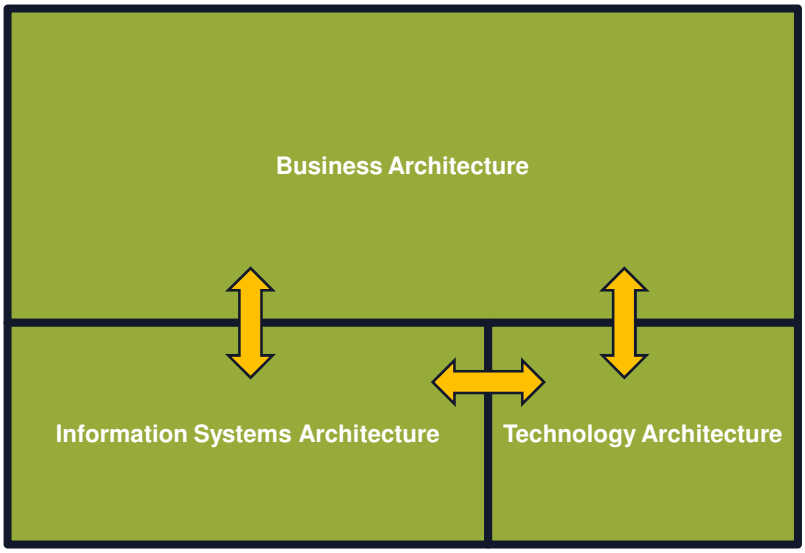
Business Change Life Cycle / Architecture Development Cycle



software AG

santam

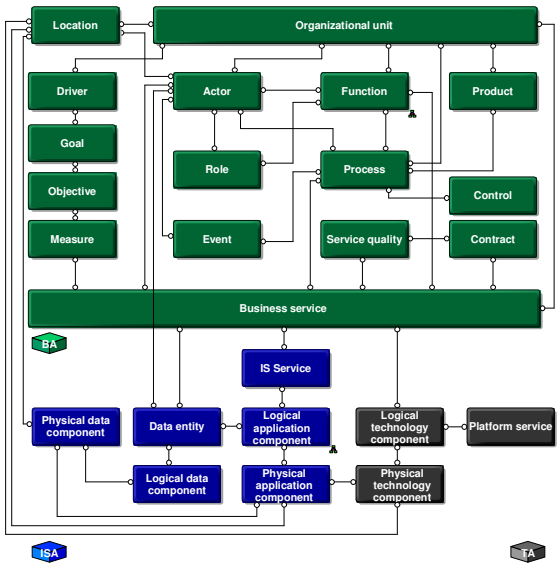
iBAS Meta Model overview



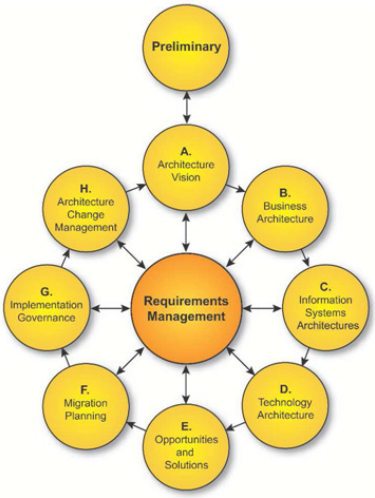
software AG

santam

iBAS Meta Model overview



iBAS Based on TOGAF ADM



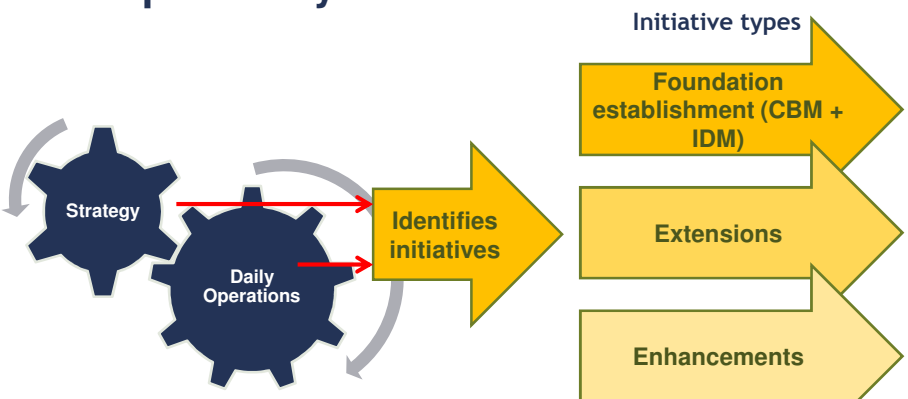
ARCHIMATE

BPMN

UML

ARIS standard

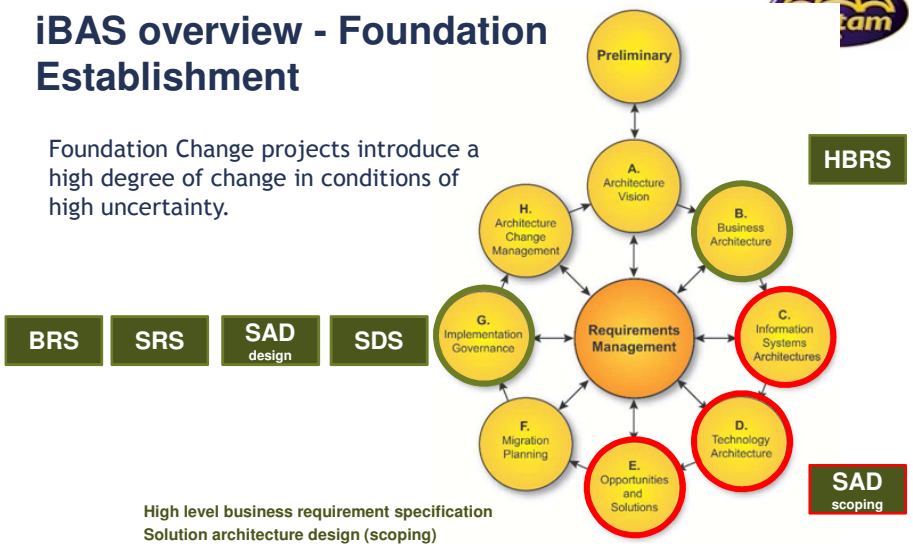
Business Change Life Cycle / Architecture Development Cycle



8th June 2011 | ProcessForum 2011

iBAS overview - Foundation Establishment

Foundation Change projects introduce a high degree of change in conditions of high uncertainty.



HBRS

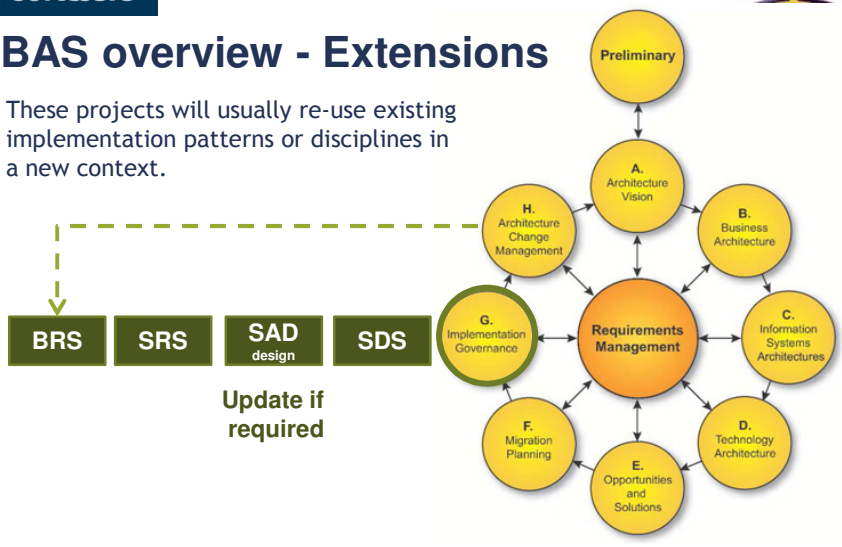
BRS | SRS | SAD design | SDS

SAD scoping

High level business requirement specification
Solution architecture design (scoping)
Business requirement specification
Software requirement specification
Solution architecture design (design)
Software design specification

iBAS overview - Extensions

These projects will usually re-use existing implementation patterns or disciplines in a new context.



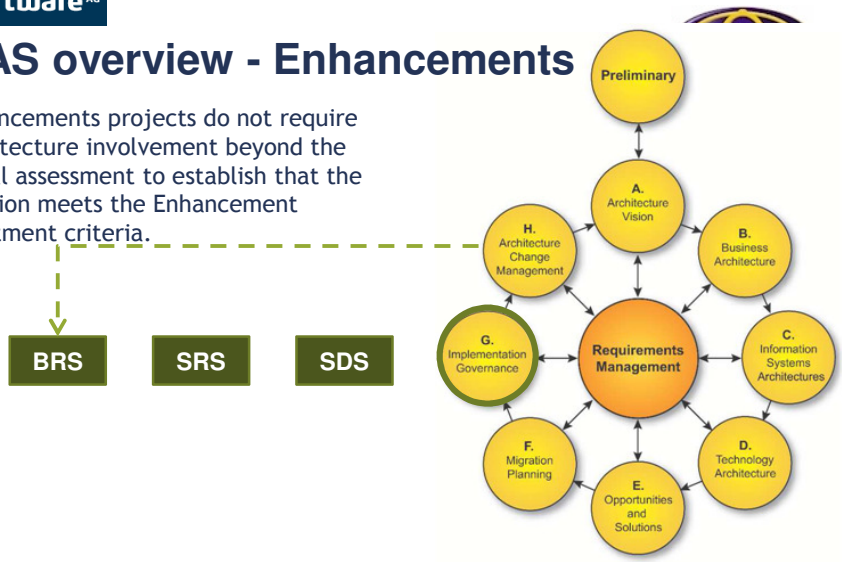
BRS | SRS | SAD design | SDS

Update if required

Business requirement specification
Software requirement specification
Solution architecture design (design)
Software design specification

iBAS overview - Enhancements

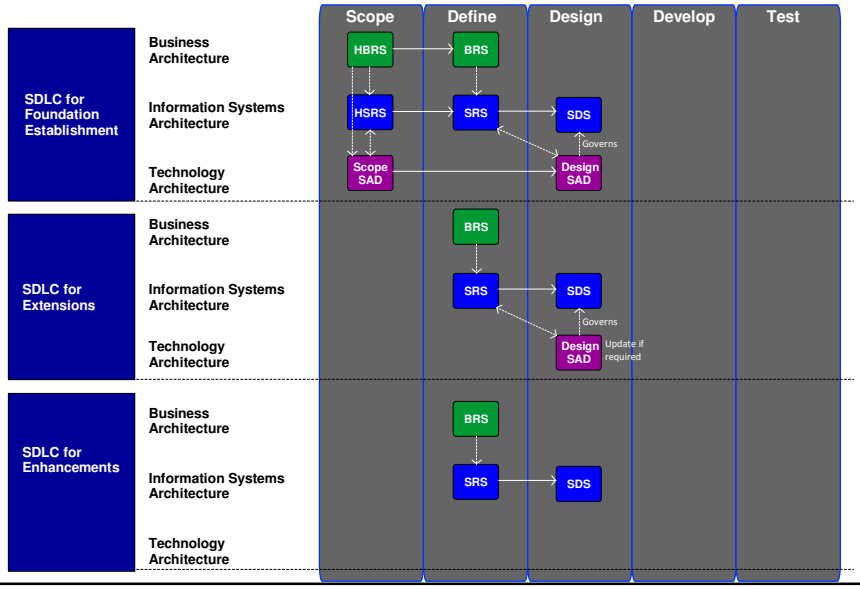
Enhancements projects do not require architecture involvement beyond the initial assessment to establish that the solution meets the Enhancement Treatment criteria.



BRS SRS SDS

Business requirement specification
Software requirement specification
Software design specification

SDLC treatments summary



iBAS deliverables – HBRS

high level business requirement specification

HBRS

1. Project Overview & Objectives
2. Project context - stakeholders cooperation
3. Project context - process scenarios in Scope
4. Glossary of terms
5. Target Operating model
   - 5.1 End to end process definition - process P1
     - 5.1.1 Business requirements - activity P1.1
     - 5.1.2 Repeat for each activity where there is business requirements
   - 5.2 Repeat for each process in scope
6. Target organizational structure & locations
7. Summary of requirements
8. Summary of measures (KPI's)
9. Summary of risks
10. Summary of unresolved issues

BA

Define the high level business requirements for the target business architecture and produce a high level business requirement specification (HBRS) which sets the scope of the project.

concept | **scope** | define | design | develop | test | deploy | maintain

iBAS deliverables – SAD

solution architecture design (sc

SAD

1. Introduction
   - 1.1 Project Overview & Objectives (reused from HBRS)
   - 1.2 SAD purpose and scope
   - 1.3 Architectural views used
2. Glossary of terms (extended from HBRS)
3. System context
   - 3.1 System context - application cooperation
   - 3.2 System stakeholders
   - 3.3 Target operating model (extended from HBRS)
   - 3.4 Summary of business, functional & non functional requirements (extended from HBRS)
   - 3.5 System scenarios
4. Architectural forces
   - 4.1 Architectural goals & critical success factors
   - 4.2 Architectural principles & decisions
5. Architectural views
   - 5.1 Baseline architecture
   - 5.2 Transition architecture
   - 5.3 Target architecture
   - 5.4 Transition roadmap
   - 5.5 Information viewpoint
   - 5.6 Deployment viewpoint

BA
ISA
TA

The scoping SAD presents a set of design principles to be used in the detailed design of the system and communicates the complete context of the solution to both business architecture and technology architecture stakeholders. It also communicates the system scope.

concept | **scope** | define | design | develop | test | deploy | maintai n

software AG

iBAS deliverables – BRS

business requirement specification

santam

BRS

1. Introduction
   - 1.1 Project Overview & Objectives (reused from HBRS)
   - 1.2 BRS purpose and scope
   - 1.3 BRS in context to process scenarios
2. Glossary of terms (extended from HBRS)
3. End to end process definition – process P1(reused from HBRS)
   - 3.1. Sub process definition SP1 (identify functional & non functional requirements) + SOA principles
   - 3.2 Repeat for each sub process
4. Screen flow & UI requirements
5. Business Service/Information Diagram
6. Organization / Actor / Role & UI access
7. Summary of requirements (extended from HBRS)
8. Summary of risks
9. Summary of unresolved issues

BA

ISA

The BRS translates business requirements into functional and non functional requirements and facilitates handover to the system analyses CoE and ensures that there is 100% traceability between the business requirements (HBRS) and functional and non functional requirements in the BRS to the SRS.

concept | scope | **define** | design | develop | test | deploy | maintain

software AG

iBAS deliverables – SRS

software requirement specification

santam

SRS

1. Introduction
   - 1.1 Project Overview & Objectives (reused from HBRS)
   - 1.2 SRS purpose and scope
   - 1.3 System context
2. Glossary of terms (extended from HBRS)
3. Software usage scenarios (use cases)
   - 3.1 Consumer use case context
     - 3.1.1 Consumer use case detail
     - 3.1.2 Consumer use case screen flow & UI requirements
     - 3.1.3 Consumer use case access rights (repeat for all relevant consumer use cases).
   - 3.2 Long running use case context
     - 3.2.1 Long running use case detail
     - 3.2.2 Long running use case screen flow & UI requirements
     - 3.2.3 Long running use case access rights (repeat for all relevant long running use cases).
   - 3.3 Short running use case context
     - 3.3.1 Short running use case detail
     - 3.3.2 Short running message requirements
4. Software logical data requirements
5. Software logical class structure (interface design model)
6. Summary of requirements & test cases
7. Summary of risks
8. Summary of unresolved issues

BA

ISA

The Software Requirements Specification (SRS) is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains non-functional (or supplementary) requirements.

concept | scope | **define** | design | develop | test | deploy | maintain

iBAS deliverables – SDS software design specification

SDS

1. Introduction
   - 1.1 Project Overview & Objectives (reused from HBRS)
   - 1.2 SRS purpose and scope
   - 1.3 System context
2. Glossary of terms (extended from HBRS)
3. Software usage scenarios (use cases)
   - 3.1 Consumer use case context
     - 3.1.1 Consumer use case detail (physical)
     - 3.1.2 Consumer use case screen flow & UI requirements (physical)
     - 3.1.3 Consumer use case access rights (repeat for all relevant consumer use cases).
   - 3.2 Long running use case context
     - 3.2.1 Long running use case detail (physical)
     - 3.2.2 Long running use case screen flow & UI requirements (physical)
     - 3.2.3 Long running use case access rights (repeat for all relevant long running use cases).
   - 3.3 Short running use case context
     - 3.3.1 Short running use case detail (physical)
     - 3.3.2 Short running message requirements (physical)
     - 3.3.3. Class collaboration
4. Software implementation design (components)
5. Software deployment design (hardware)

SDS

6. Software physical data requirements
7. Software physical class structure (interface design model)
8. Summary of requirements & test cases
9. Summary of risks
10. Summary of unresolved issues

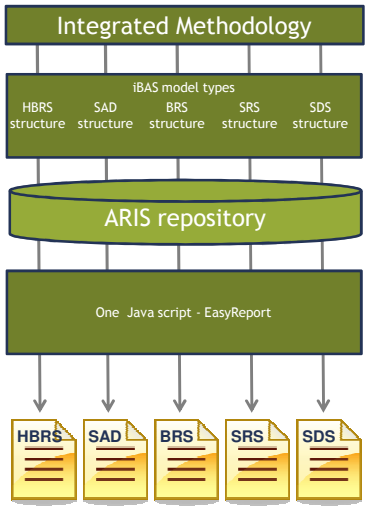
The software design specification (SDS) is a written description of a software product, that a software designer writes in order to give a software development team an overall guidance of the architecture of the software project.

ISA

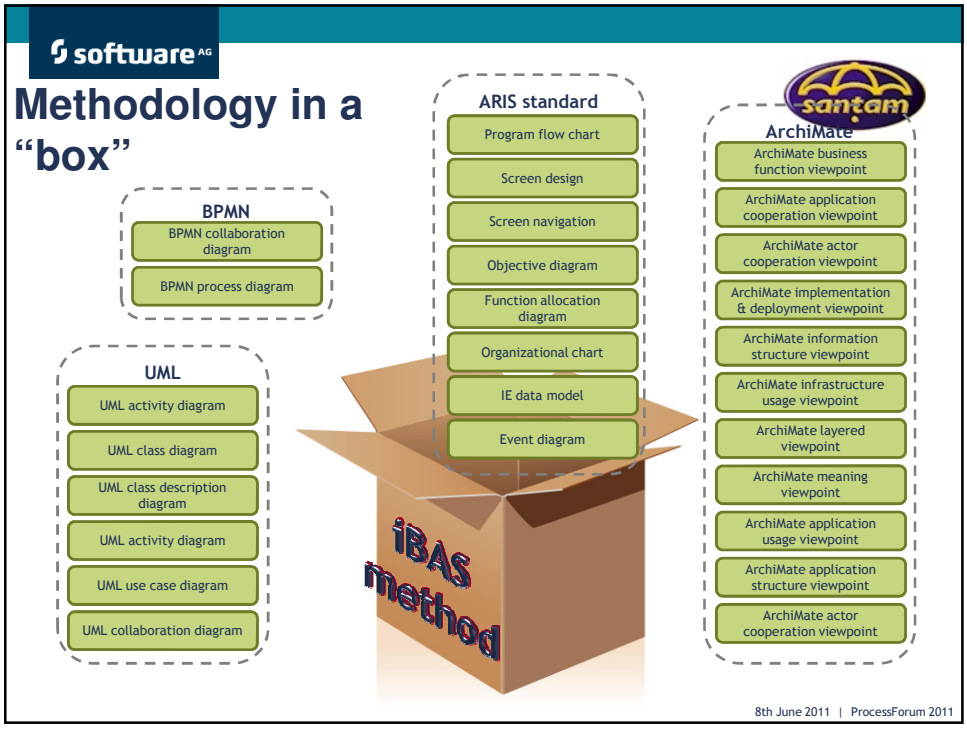
TA

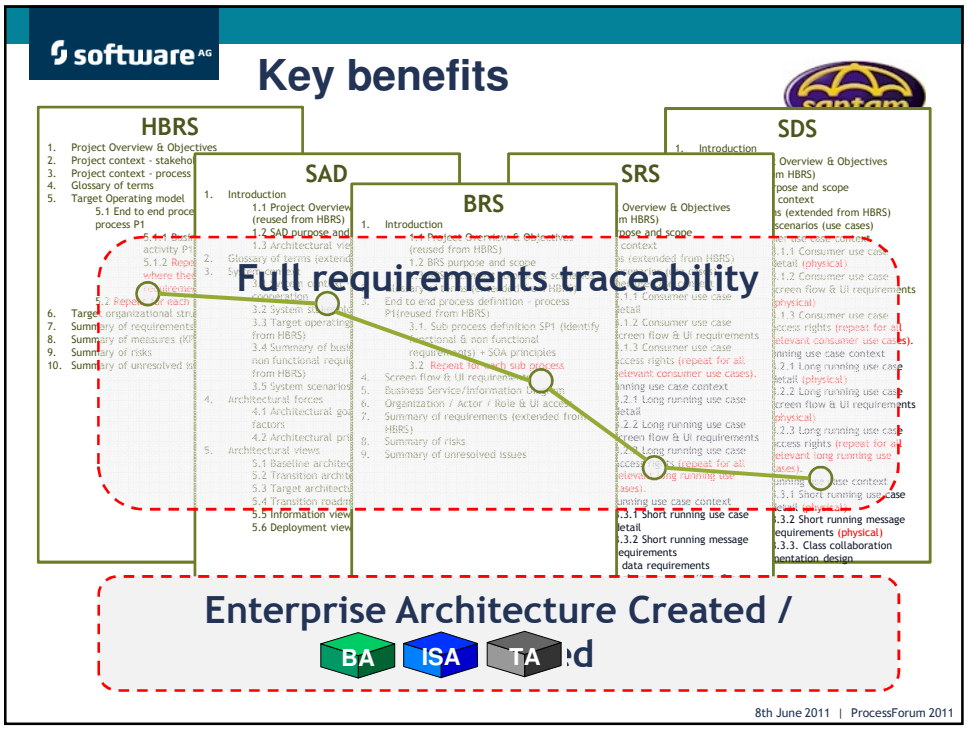
concept | scope | define | **design** | develop | test | deploy | maintain

Methodology in a "box"



e-specifications


software AG
Methodology in a "box"
BPMN
BPMN collaboration diagram
BPMN process diagram
UML
UML activity diagram
UML class diagram
UML class description diagram
UML activity diagram
UML use case diagram
UML collaboration diagram
ARIS standard
Program flow chart
Screen design
Screen navigation
Objective diagram
Function allocation diagram
Organizational chart
IE data model
Event diagram
iBAS method
ArchiMate
ArchiMate business function viewpoint
ArchiMate application cooperation viewpoint
ArchiMate actor cooperation viewpoint
ArchiMate implementation & deployment viewpoint
ArchiMate information structure viewpoint
ArchiMate infrastructure usage viewpoint
ArchiMate layered viewpoint
ArchiMate meaning viewpoint
ArchiMate application usage viewpoint
ArchiMate application structure viewpoint
ArchiMate actor cooperation viewpoint
8th June 2011 | ProcessForum 2011


software AG
Key benefits
HBRS
SAD
BRS
SRS
SDS
Full requirements traceability
Enterprise Architecture Created /
BA
ISA
TA
8th June 2011 | ProcessForum 2011


software AG
ProcessForum
Changing the World - One Process at a Time
Thank You!
8th June 2011 | ProcessForum 2011