



Software Development Life-cycle Hygiene with Message Broker in end-to-end SOA

Tips for building and maintaining a working SOA implementation in MB

Stuart Smith - Consultant, Smart421
ssmith@smart421.com

- How to architect a SOA
- How to implement/realise a SOA with MB

- Suggestions/Tips for ensuring your MB based SOA 'works'
- Discussion of the skills your developers will need
- A look at why hygiene is so important to get it to 'work'

- Smart421 splits it into two parts:
- the glue and base capabilities (aka ESB)
- the re-usable services

- It's the standards and policies for exposing and consuming the **first class interface** of a component being integrated.
- And if it has good hygiene, it also has a good story for:
 - transactionality
 - security
 - logging, monitoring, error handling
 - QoS
 - easy reuse
 - low cost of change

- Fundamental shift in ownership from the designer/developer owned 'interface spec' to the business architect owned **services**
- ESB is no longer just a term used for 'plumbing' : it is an enterprise app in its own right...
- Business services provide high value reuse across the organisation
- Technical services provide a common set of capabilities to all those 'higher' services (e.g. logging, security, monitoring, SLAs. etc)

- A fully realised ESB
- Lots of reusable (**and reused!!**) services
- Many services are business focused
- Services exhibit tiering
- Services easily enhanced & extended with high confidence
- Quick to build, quick to change

- Business focused services
- Open standards (SOAP, etc)
- Designed from outset for reuse
- OO patterns used (e.g. separation of concerns)
- Designed from outset for testability
- Test Driven Development/CI

- Business focused services
- Open standards (SOAP, etc)
- Designed from outset for reuse
- OO patterns used (e.g. separation of concerns)
- Designed from outset for testability
- Test Driven Development/CI

- MB is for when you can't do it with just MQ
- Some transforms, the odd protocol transfer - but basically point to point plumbing
- MB devs might be more *admins* than *software engineers*
- Admin led approach leads to little, if any, design or documentation

it ain't heart
Surgery!



- “bodged together until the two apps are ‘*connected*’ ” ?
- Probably no unit testing to speak of...
- MB support team own the ‘tin and wires’ and the bit of plumbing: idea of service level ownership an anathema

old Skool
MB
SDLC \neq a "working"
SOA...

- on/off ramp
- validation
- routing
- transformation
- error handling
- logging
- auditing
- monitoring
- SLA enforcement
- security
- etc, etc, etc



- Service provider has ownership of service.
- Creation and Enforcement of a set of behaviour:
 - validation of messages (schema, complex, partial)
 - integration with ESB core services
 - uses other services
 - uses other ESB shared components (canonical data format, frameworks, libraries)
 - quick to build, quick to adapt
 - exist in a design time repository



The Story so far...

The ESB is the **heart** of the organisation.

The business services running on it, **are** your organisation.

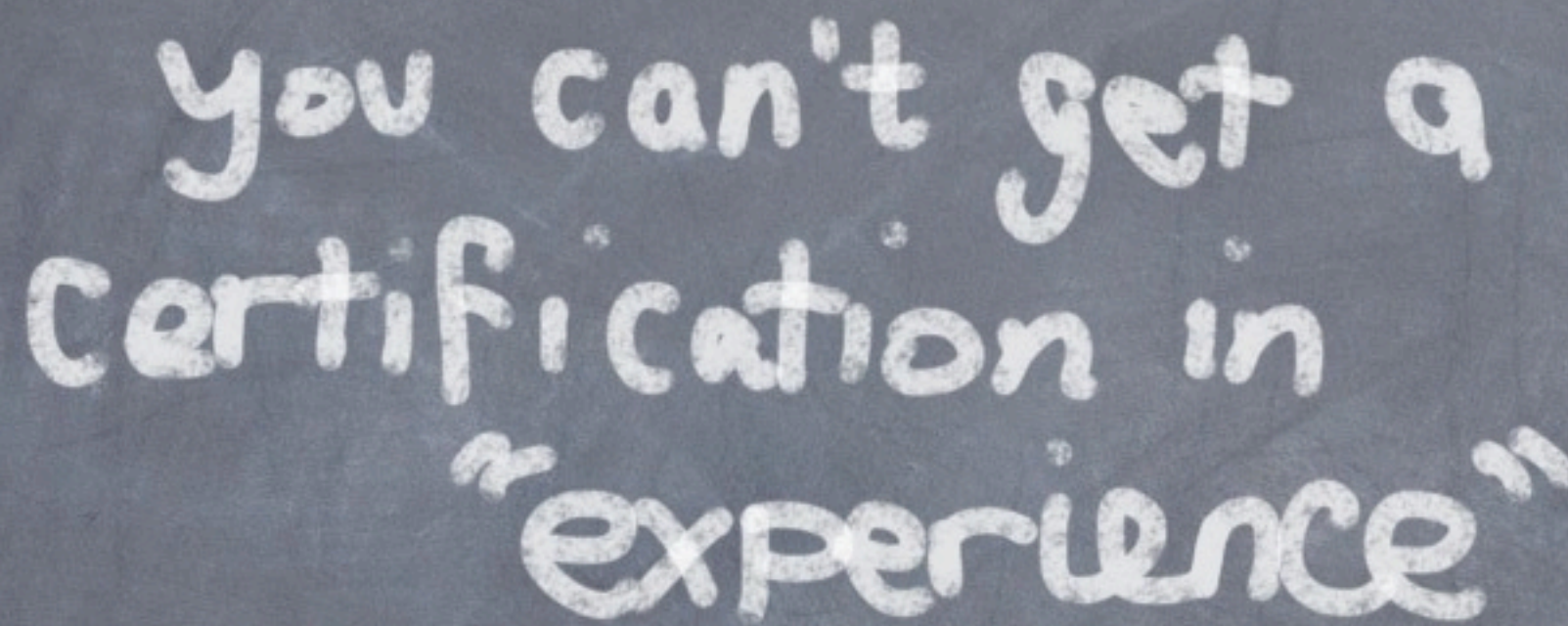
Building and Maintaining it, should be a selection of your brightest and most experienced software developers.

IT IS HEART
SURGERY!

- Business focused services
- Open standards (SOAP, etc)
- Designed from outset for reuse
- OO patterns used (e.g. separation of concerns)
- Designed from outset for testability
- Test Driven Development/CI

- Senior Software Engineer
- Excellent OO Knowledge and Experience
- In Depth knowledge of software patterns
- Lots of enterprise level experience (probably including lots of different technologies)
- Interface design experience
- WAS-ESB Knowledge and Experience

- Senior Software Engineer
- Excellent OO Knowledge and Experience
- In Depth knowledge of software patterns
- Lots of enterprise level experience (probably including lots of different technologies)
- Interface design experience
- MB Knowledge and Experience



you can't get a
certification in
"experience"



Skill Summary

Existing (even senior) MB developers are just as likely to be lacking required core skills as a junior developer

Retraining/Restaffing, External Consultancy will almost certainly be required to get your ESB team in shape.



We've got the right people - what now ?

they will tell you !

Output	Role	Inputs
Solution Design	Solution Designer	service contracts, app designs, requirements, ESB team input
Service Contact(s), Service Design(s)	Service Designer, Lead MB ESB team	solution design, existing service contracts, existing service designs
TDD	MB ESB Team	Service Contracts, Service Designs

- schemas
- security
- pre/post conditions
- error handling
- configuration
- non-functionals
- interface versioning
- SLAs, QoS

- designed for loose coupling (inc time dimension: e.g. asynchronous or datagram message patterns)
- sequence diagrams covering as many paths as required
- more detailed pre/posts (inside the black box)
- detailed error handling in MB flows, error codes, etc
- deployment characteristics
- transport choices, details associated with them (e.g. queue names, etc)
- mapping specifications
- validation tasks
- 'Last Mile' designed here too - same principles applied here. interface specified, transports, etc

- TDD can begin alongside service contract/design stage, or traditional waterfall.
- Code Repository used, code reviews enforce detailed coding standards and design patterns.
- Build tests for every pre/post condition, every error condition, all the non-functionals
- AFTER you write the test, write the code
- Repeat until full coverage
- Wrap all these tests into CI and: easy service enhancement, refactoring with confidence, quick to respond to business.
- Regression tests suites
- Apply TDD to 'last mile' interfaces to applications too - not just the first class interface

- Better designed code
- Higher reuse
- Lower bugs
- Easier maintenance
- Easier refactoring with high confidence
- Almost a prerequisite for a 'working' SOA



Smart421

www.smart421.com

- Part of the KCOM Group (£400M, 2200 people)
- IBM Premier Business Partner
- IBM Tier 1 WebSphere Integration Services Partner
- Smart421 SOA Consultancy
- Large DataPower and Message Broker practice
- Experience of numerous WebSphere projects

