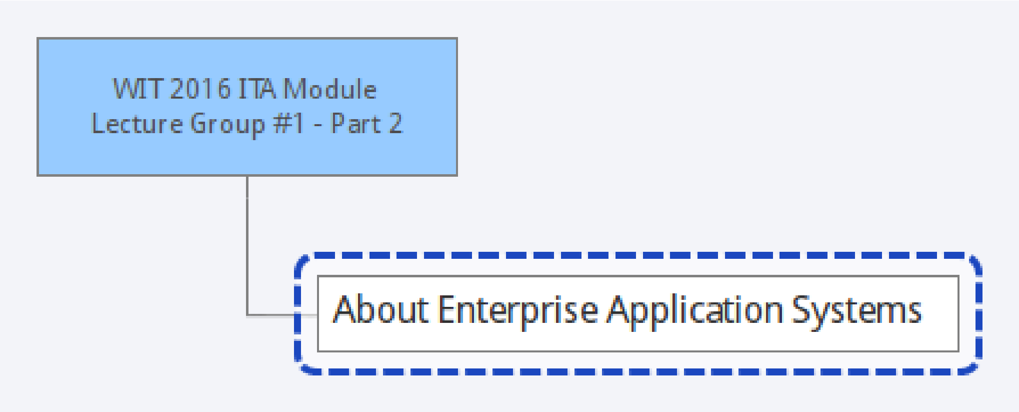
WIT 2016 ITA Module

Lecture Group #1 - Part 2
Enterprise Application Systems (EAS)

Lecture Group #1 - Part 2



WIT 2016 ITA Module
Lecture Group #1 - Part 2

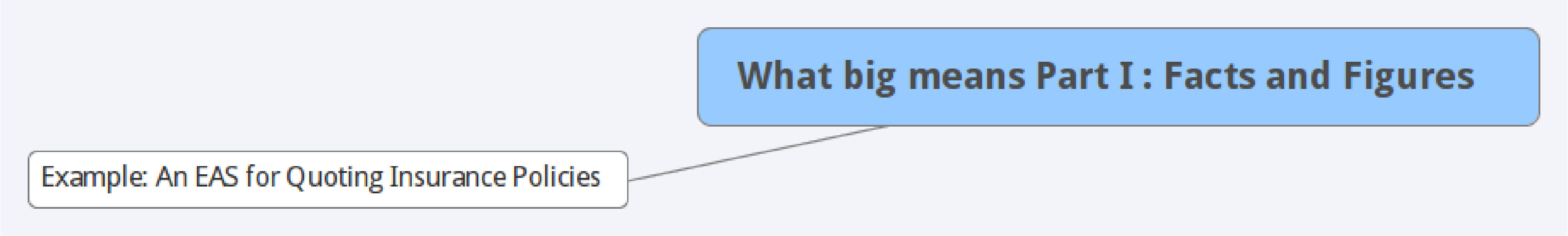
About Enterprise Application Systems

What big means: Facts and Figures

We hear a lot about large-scale application deployed in corporations

...but what does “big” mean?

The following present some real-life data-points gathered from interviewing the product-owner of a software platform allowing an insurance company to provide quotes for insurance policies to its prospect customers.



What big means Part I : Facts and Figures

Example: An EAS for Quoting Insurance Policies

Business Operations standpoint

Maintained by a team of 30 developers (+ 20 outsourced)

Developments teams spread across 3 countries

Tested by by a team of 25 quality analysts

Supported by 10 production system analysts

Application standpoint

Composed ~=300 visible Web UI screens

Delivered to 1850 active users, up to ~=3500

Supporting 15-20 business processes

Designed using a layered architectural style

Relies on 10 Business Rules Engines

Running on J2EE platform / EJB 3

~=20 EAR deployable components each

Interoperability with 14 other applications via an ESB

Communications mostly via message queuing (JMS), a few SOAP WS

External communications using EDI, mostly XML, some JSON

Internal communications using Value Objects (RMI)

Data standpoint

Data access mainly via ORM, some POJO

5 MS SQL 2015 databases in total supporting the system

756 tables in core RDBMS

1 TB after database compression on five core tables

Largest table has 1 billion rows

Significant amount of stored procedures logic

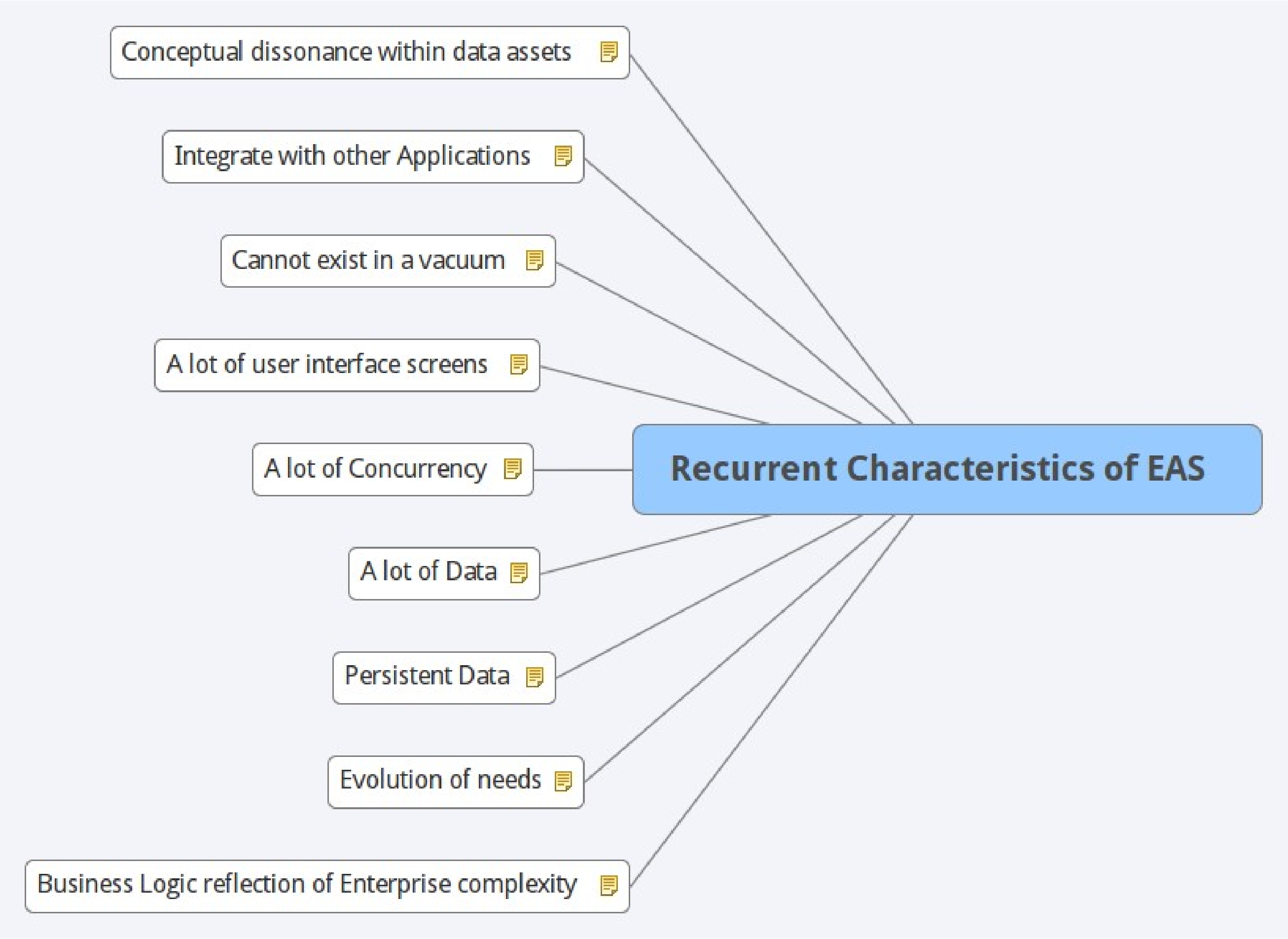
Infrastructure Operations standpoint

Deployed in AIX Virtual Machines - Weblogic 11g, java 7

3 weblogic server instances

Spread across 14 clusters

Characteristics of Enterprise Applications



Recurrent Characteristics of EAS

- Conceptual dissonance within data assets
- Integrate with other Applications
- Cannot exist in a vacuum
- A lot of user interface screens
- A lot of Concurrency
- A lot of Data
- Persistent Data
- Evolution of needs
- Business Logic reflection of Enterprise complexity

A reflection of Enterprise complexity

Business pressure (pursuit of efficiency/profitability force unexpected conditions that push ESS to often interact with each other in unexpected ways.

Many one-off special cases is what can easily lead to the complex business "illogic" that makes business software so difficult.

In stark contrast with systems software in computing sciences: highly logical and scientific.

Constant evolution of needs

Business problem to solve for changes shape faster than applications take to be built.

Business opportunities systematically emerge in mid-flight of 80 IT projects/year running concurrently.

Data as business asset

Data is required to be around for months and usually years (ex. auditing, analytics)

.. during which there will be many changes in the applications that consume/produce it.

During lifetime of an application, changes to the structure of the underlying data...

... requiring ‘evolution’ of data based on older schemes.

Some EAS may even outlive hardware, operating systems and compilers that created it.

A lot of Data

A moderate system will have over several GB of data organized in tens of millions of records...

...replacing older systems used indexed file structures such as IBM's VSAM and ISAM.

Use mostly relational databases...

... although other forms of data persistence are beginning to be an option.

The design and feeding of these databases has turned into a sub-profession of its own.

A lot of Concurrency

For many systems this may be less than a hundred people...

... however, for Business critical or Web-based systems, it may be thousands or hundreds of thousands.

Major issues arise in ensuring that all users can access the system properly....

... with transaction managers required to ensure two users don't access the same data concurrently at the same time in a way that causes errors (ex. long running queries causing table locking, data integrity issues).

A lot of User Interface screens

Potentially hundreds of distinct screens...

...with role-based security, many user groups, having highly varying levels of technical expertise.

Information has to be presented in many different ways for different purposes....

...and synchronicity of data presented to all users, in the context of the process they act upon, must be guaranteed.

Applications rarely exist in a vacuum

Enterprises rarely have the luxury to re-write existing applications, convert databases, nor like to re-invent the wheel.

New systems must collaborate with older systems to fulfill a task.

Relationships between systems take many shapes and forms.

Dependencies between applications only make sense in the context of what business users need to achieve.

Information Technology is governed

There's little incentive now for large technology corporations to consolidate except when the technology is dying out.

Enterprise applications may be built at different times with different technologies

...and with different collaboration mechanisms (COBOL data files, legacy messaging systems, etc.)

Occasionally the enterprise will attempt to integrate its different systems using a common communication technology

...but often not completed, always work in progress, performed in yearly increments

...leaving several different unified integration schemes in place at once.

Conceptual dissonance within data assets

As the business grow, miscommunication, misinterpretation, misrepresentations of data definitions occur.

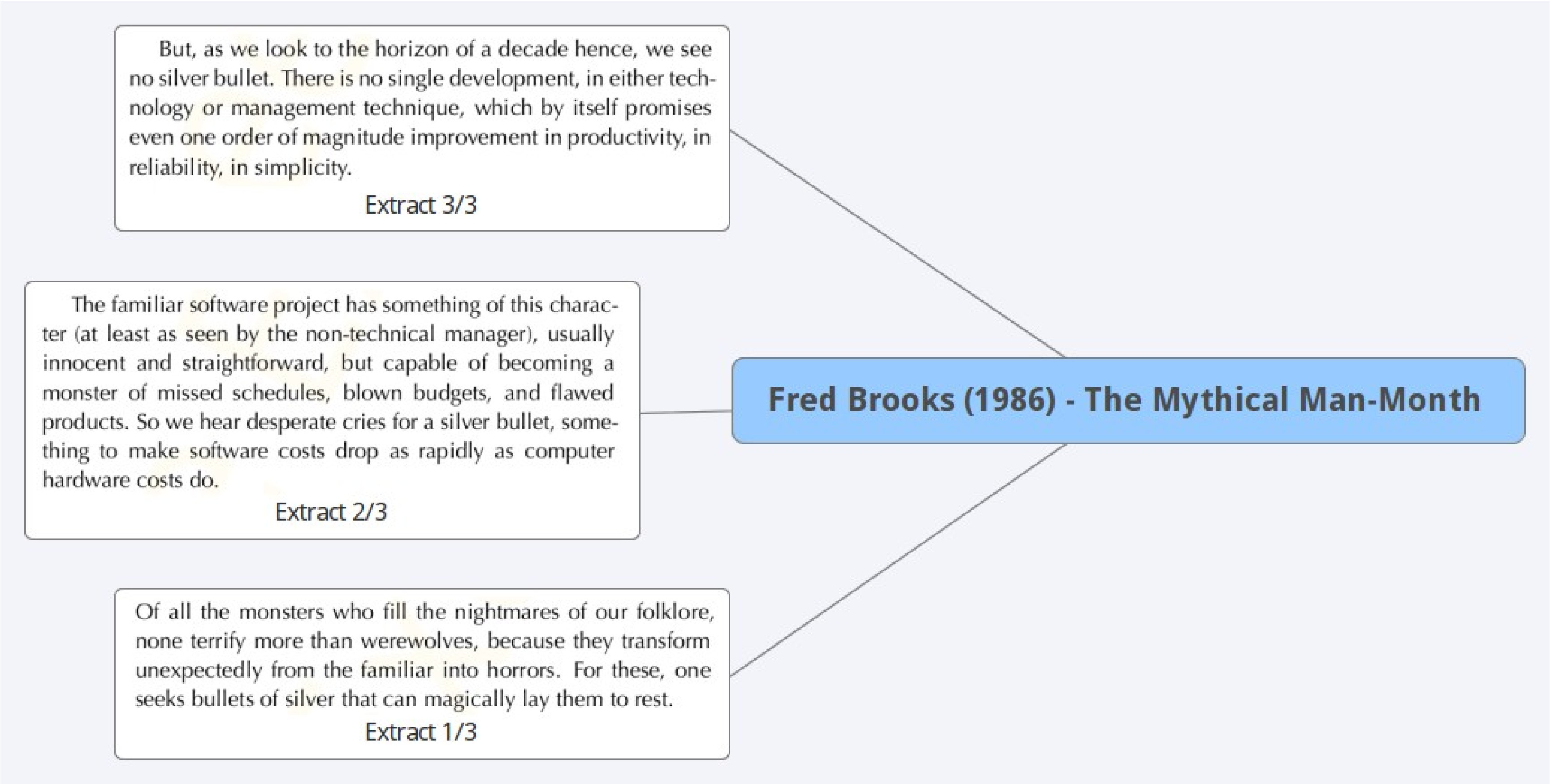
Business definitions differ by business functions, due to differences in business process

...creating conceptual dissonance within enterprise data assets (ex. when does a prospect customer becomes a 'customer'?, at what point of the distribution process do we record a 'sale'?).

One division of the company may think a customer is someone with whom it has a current agreement; another division also counts those that had a contract but don't any longer.

As a result, data has to be constantly read, adapted, and written in all sorts of different syntactic and semantic formats, resolved in a master data management data repository.

No Silver Bullet



Fred Brooks (1986) - The Mythical Man-Month

But, as we look to the horizon of a decade hence, we see no silver bullet. There is no single development, in either technology or management technique, which by itself promises even one order of magnitude improvement in productivity, in reliability, in simplicity.

Extract 3/3

The familiar software project has something of this character (at least as seen by the non-technical manager), usually innocent and straightforward, but capable of becoming a monster of missed schedules, blown budgets, and flawed products. So we hear desperate cries for a silver bullet, something to make software costs drop as rapidly as computer hardware costs do.

Extract 2/3

Of all the monsters who fill the nightmares of our folklore, none terrify more than werewolves, because they transform unexpectedly from the familiar into horrors. For these, one seeks bullets of silver that can magically lay them to rest.

Extract 1/3