MAP-WayPointer

A Smart, Agile Method for Large Scale Enterprise Application Development

smita.ghaisas@tcs.com

Agile approach

Do not scale up to large projects with large teams

Require customer to be present on-site- not suitable for outsourcing assignments

No traceability to requirements

Process intensive approach

Heavy weight detailed documentation

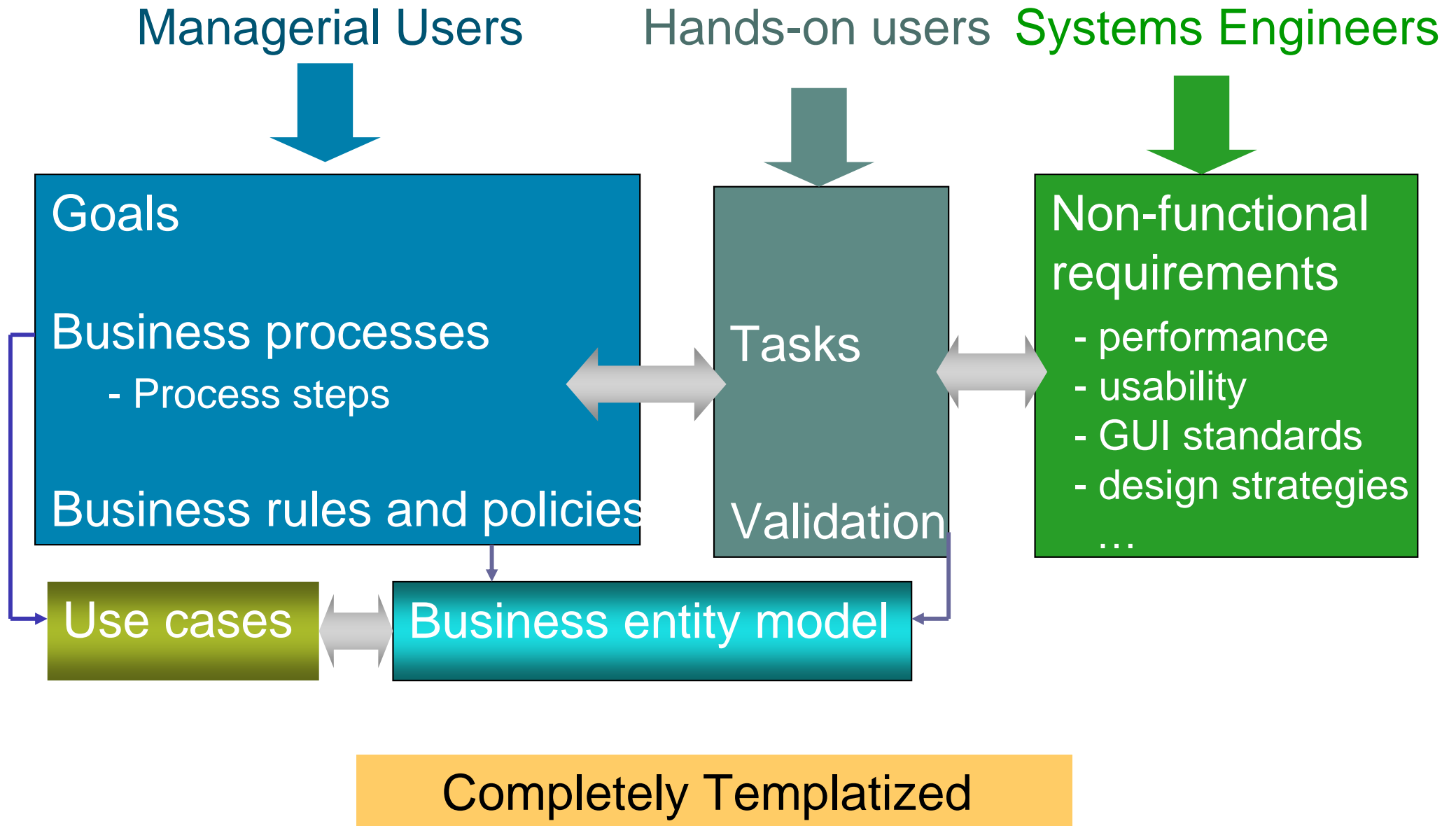Documents go quickly out of sync with implementations

Vagaries with artifacts to be delivered and their properties

Need a *smart, agile* method for large scale development

The MAPAGILE approach…

- Designed to address a typical on-site- off-shore scenario
- Provides light-weight templates to capture inputs from different types of stakeholders
- Incorporates many cross checks for their consistency, completeness and correctness
- Ensures business alignment of stakeholders' expectations
- Separates functional and technical concerns
- Addresses core requirements of different types in the first iteration
- Provides a mechanism to convert captured inputs into SDLC specific artifacts
- Clearly states verification and validation criteria for artifacts

The MAPAGILE snapshot

Managerial Users   **Hands-on users**   **Systems Engineers**

Goals

Business processes

  - Process steps

Business rules and policies

Tasks

Validation

Non-functional requirements

- performance
- usability
- GUI standards
- design strategies
…

Use cases   ⟷   **Business entity model**

Completely Templatized

How do we start?

- Understand organizational context
  - What role will the proposed application play?

- Understand business goals
  - What business goals are expected to be met?

- Identify stakeholders
  - Managerial users and hands-on users

- Understand expectations of managerial users
  - How do they plan to use this application?

- Ensure goal alignment of expectations
  - Which of the business goals will be met as a result of meeting these expectations?

The first iteration…
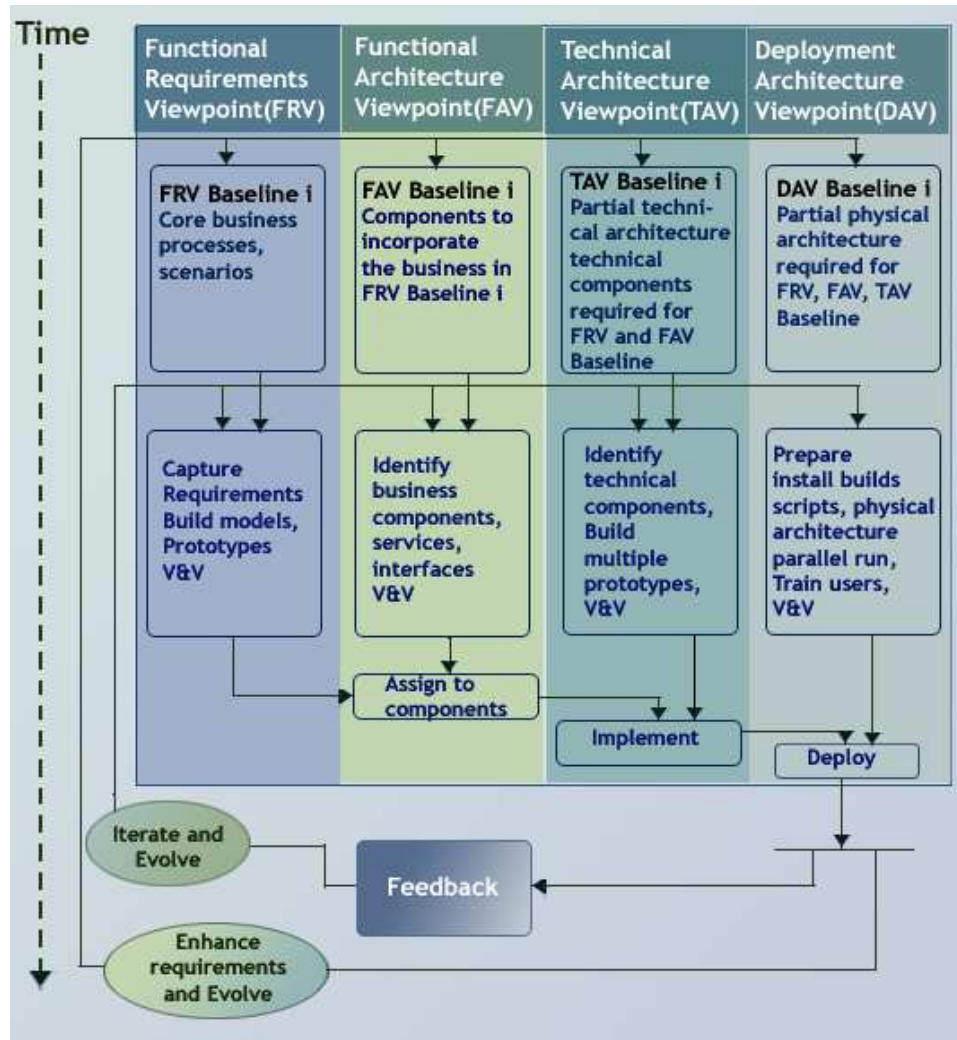
- Core business processes
  - Functional requirements
- Functional scope boundaries spanned by processes- Components
  - Modularization/ Segmentation related requirements
- System response time for critical transactions, peak load, concurrency…
  - High priority non-functional requirements
- Issues such as availability of application, support , release plan…
  - Deployment related requirements

Artifacts- some examples

Light- weight use cases

- Process that owns the use case
- Actors
- Stimulus
- Pre and Post conditions
- Inputs
- Steps and validations
- Output
- Associations

Business entity model

- Consistent with use case specification
- Rules and policies to derive associations and cardinalities of class model
- Clear guidelines and templates for class model, GUI design, database design

MAPAGILE User Base

- 35+ users across TCS delivery centers
  - Includes model-based, ERP-based, code-driven projects

- Plug in to SAP solutions
  - MAPAGILE as a methodology for the next release of the solution

- Adopted as an organizational method by customers in UK, Ireland- Insurance and banking

Smart MAPAGILE

We need

– Active and intelligent guidance with activities

– Faster, easier learning

– Automated review process

Making knowledge explicit and available just-in-time

MAP-WayPointer- a "paired" experience

- Enables a correct sequence of activities
- Watches artifacts and points out inconsistencies
- Watches out for missing artifacts
- Provides a number of focus questions and examples
- Provides a user-friendly interface to capture stakeholder inputs, display reference literature and track analysis artifact details.
- Provides a report on detected inconsistencies
- Provides a just-in-time review- as you work with MAPAGILE
- Improves productivity by reducing learning time, reviewing time,number of review cycles
- Improves quality of your deliverables by providing an in-built review mechanism