



# Behind the scenes

Spatial Dimension's Development, Test & Support methodologies

An MS Group Business

# Background

---

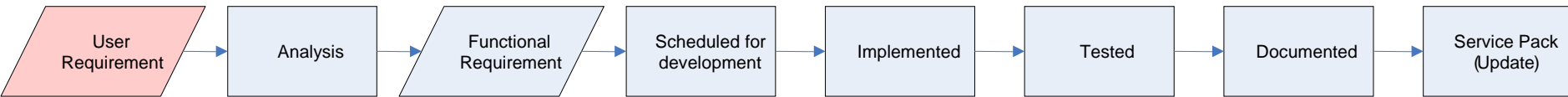
- Since the last User Conference
  - Number of installations of FlexiCadastre worldwide has doubled
  - Staff Compliment has more than doubled
  
- Agile software development methodology had to be streamlined for a bigger team and client base, in order to maintain our level of excellence
  - Small team, roles overlap
  - Bigger team
    - roles defined more clearly
    - processes more structured

# Methodology overview (the vision)

---

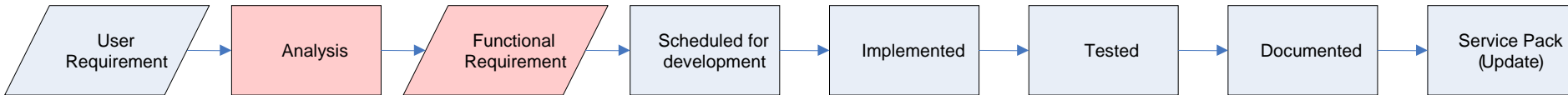
- Evaluated existing methodology
- Identified areas to streamline
- Defined new processes
- Looked for a tool to assist with managing the processes
- FogBugz used as communication tool

# User Requirements



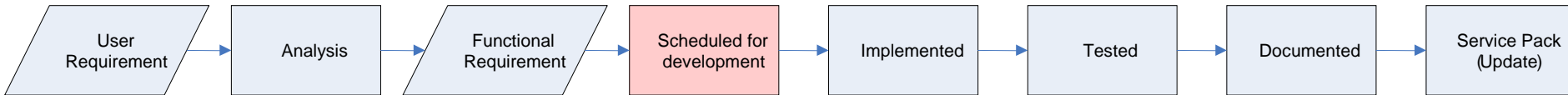
- Professional services staff (Project Managers and Business Analysts) log user requirements per client project in FogBugz
- Progress on development available on project level

# Analysis and Design



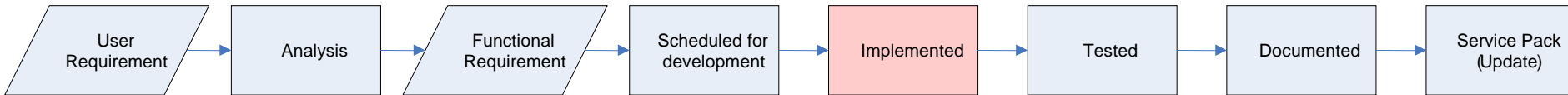
- Analyst receives user requirements through FogBugz
- Analyses:
  - Power vs. Usability (e.g. action series setup)
  - Vision vs. User requirement (e.g. mine level data management)
- User requirement is converted into one or more functional requirements that can be implemented by developers
- Analyst includes time estimation for development

# Scheduled for implementation



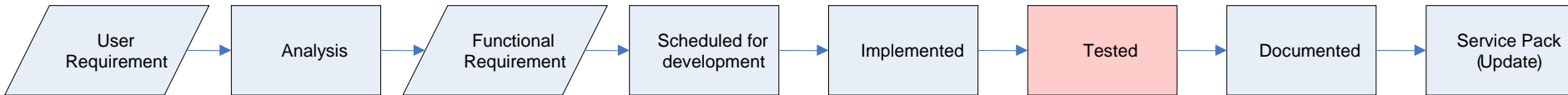
- Release Manager schedules functional requirement(s) to be developed as part of a shipping version
- The overall vision for FlexiCadastre as well as client expectations will drive the shipping dates and content (ArcGIS Server, Multilingual Support, Interface changes)

# Implementation



- Developer receives notification to implement functional requirements, prioritised within the shipping version
- Developer re-assesses time estimation and adjusts if necessary
- Developer implements according to analysis

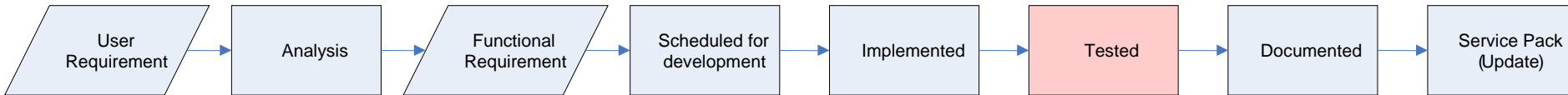
# Testing - Developer



- Once implemented, the developer does unit level testing to ensure the new code works, and that it does not affect existing code
- Testing script becomes part of the of unit test script repository

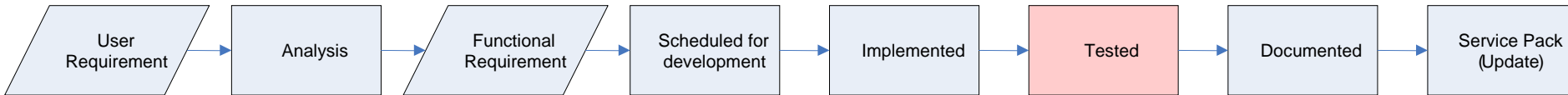


# Testing - Tester



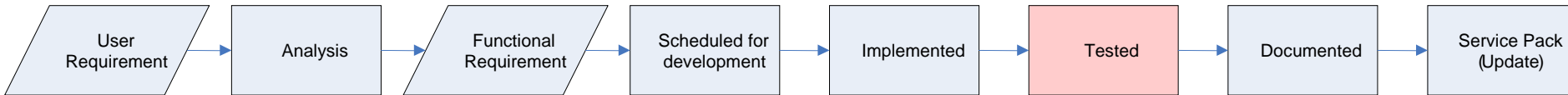
- The tester receives notification that the unit tests were successful and that the newly implemented functionality is ready for interface testing
- An interface test script is developed (using Compuware's TestPartner)
- Once the script runs successfully, it becomes part of the interface test script repository
- Every night all scripts in the repository are re-run to ensure that any bugs introduced yesterday, can be fixed today.

# Testing – Professional Services



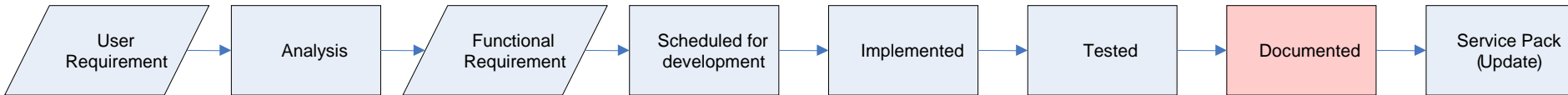
- All project manager whose clients requested the developed functionality, receives notification that is has been implemented.
- The project managers test functionality against the original user requirements

# Testing – Challenges



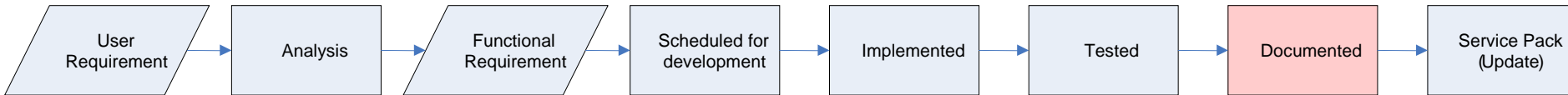
- Moving from manual testing to automated testing
- Script repositories have to be built up to cover existing functionality
- Vast amount of different type of tests that could be included in the test plan
- Identified the first ones to implement (Unit level, Interface)
- Evaluating tools to do performance analysis

# Documentation



- Documenter gets a notification to document new functionality
  - Online Help file
  - Release notes for the new piece of functionality added to the set of release notes for the shipping version

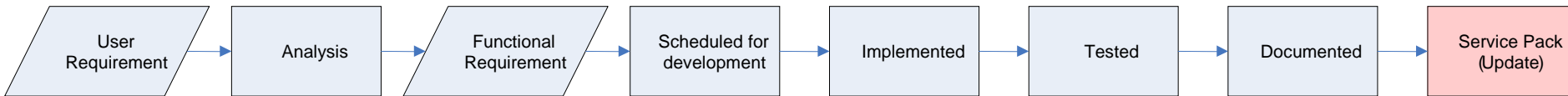
# Documentation - strategy



- **Online help files**

- Standard, condensed help files (step-by-step)
- FlexiCadastré terminology and vision (Power)
- Client specific implementation details (Usability)

# Preparation of Service Packs



- Once all functionality and bug fixes that were planned for a shipping version has completed the development cycle, the version is ready to be shipped.
- While a service pack is being built, the client is informed of the new version that is available (including release notes).
- A date is set with the client for the planned update.
- Every shipping version will have a set of release notes, including a list of new functionality and details of fixed bugs (not black box or Jack-in-the-box)

# Client side Quality Assurance

---

## QA Environment vs. Production Environment

- QA Environment updated on an agreed date
- Client has opportunity to test (agreed timeframe)
- A date is set with the client for the planned update on production environment
- Production environment updated on agreed date

# Version Control

---

## **Shipping versions could be either Development or Release versions**

- Common set of underlying computer code
- Clients in different phases of implementation e.g. Prototype, Pre-production, Production
- Once in production, running on release version
- Before that, development version



# Communication & Support

---

- Remote Access to client servers from Cape Town
- User discussion forums (FogBugz)
- Global partnerships

# Testing - Demo

---

## Compuware TestPartner Demo

# Any questions?

---