

*Organisations face an increasingly complex environment when making decisions about business applications*



## **Business Pressures**

- Focus on cost reduction
- Increasing pace of change
- Need to create and sustain competitive advantage
  - Importance of customer satisfaction and loyalty
    - Demand for critical business information
- Compliance with industry regulations & standards

# Challenges we faced



# Our aspirations



# Finding the right way of working

- Less than satisfactory experience with RUP & other methods
  - *A desire to move away from One-Size-Fits-All methods*
  - *A desire for more Agile approaches*
- *A need for a scalable methodology*
- *A need to conform to CMMI*

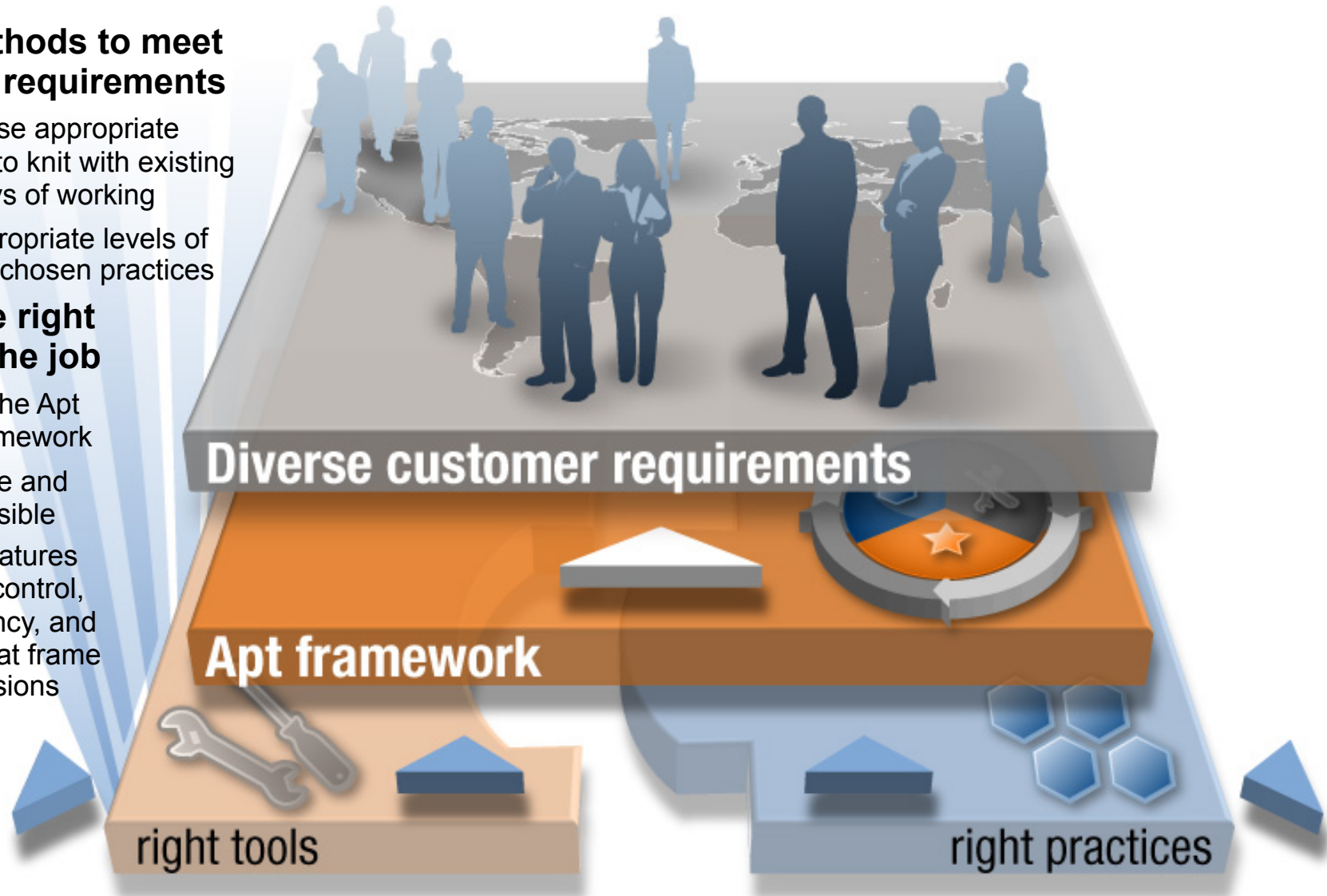


# A meeting of minds 2006

- New approach
- No monolithic process
- Aspect orientated view
- Core underlying structure
- Just enough process

# Just enough way of working

- **Meeting the diverse customer requirements to method**
- **Tailor methods to meet customer requirements**
  - Choose appropriate practices to knit with existing ways of working
  - Set appropriate levels of rigor for chosen practices
- **Select the right tools for the job**
  - Within the Apt tools framework
  - Flexible and extensible
  - Core features deliver control, consistency, and quality that frame extensions



# Apt modern development techniques

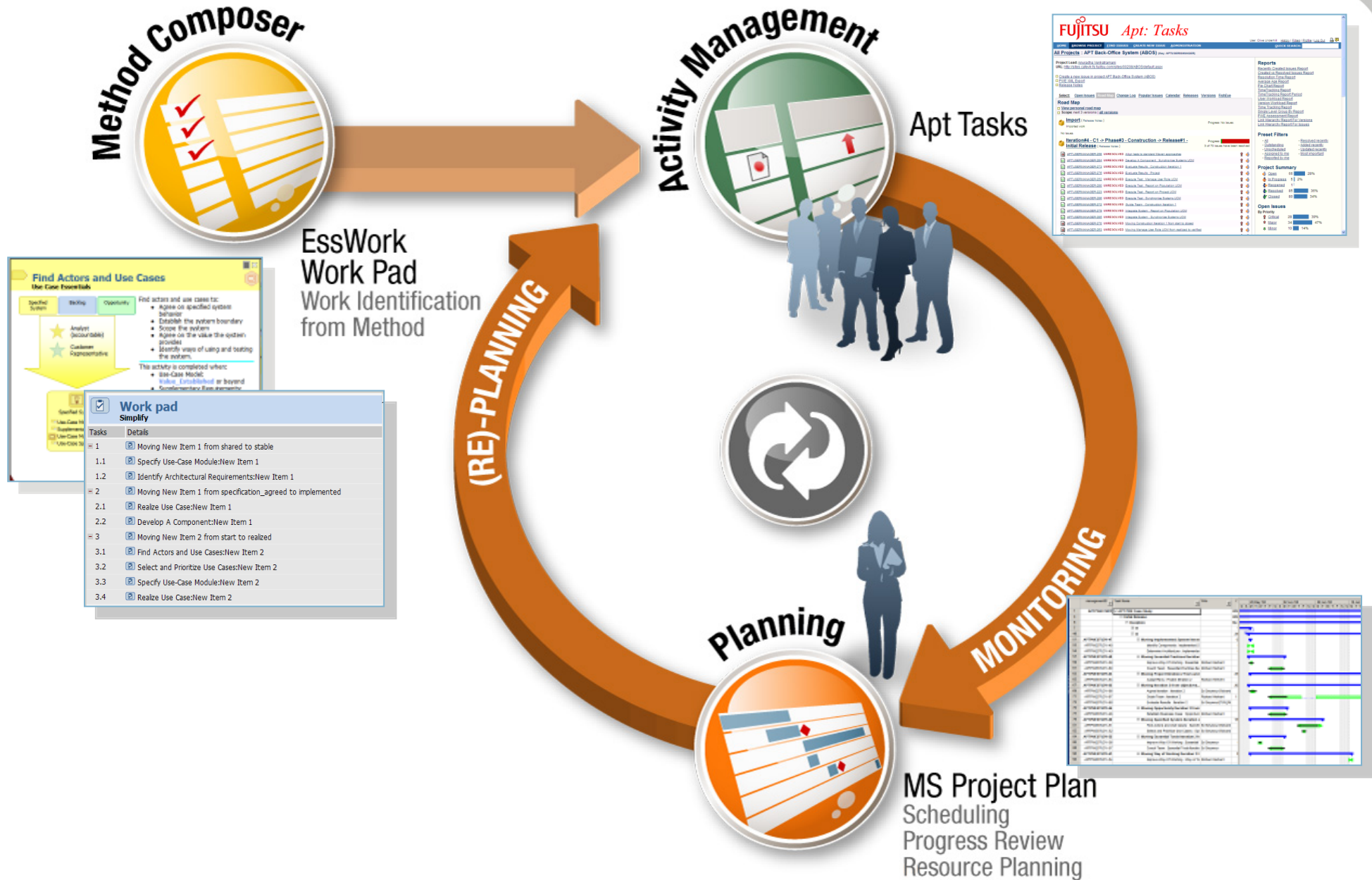


# *Apt* improves consistency & collaboration





# Apt Planning & activity management



## ■ Planning an iteration or work package

- Team agree a set of goals
- Automatic generation of Activities to achieve goals using EssWork
- Publishing of Activities for tracking
- Scheduling with round-trip integration with MS Project & JIRA

The screenshot shows a 'Use-Case Module Gameboard Demo' interface. It features a table with columns for 'Item', 'Start', 'Scope', 'Specification Agreed', 'Realized', 'Implemented', and 'Verified'. Two rows are visible: 'Update Status - Reply + Mention' and 'Follow friend - Basic Path'. Callouts provide context: 'Items of concern' points to the 'Item' column; 'Agreed current status' points to a yellow icon in the 'Scope' column; 'Target status sets the Goal' points to a red target icon in the 'Verified' column.

Item	Start	Scope	Specification Agreed	Realized	Implemented	Verified
Update Status - Reply + Mention						
Follow friend - Basic Path						

# Apt Automatic Generation of Activities

A Goal

Goal's Activities

Work pad Demo		
Tasks	Details	
- 1	<input checked="" type="checkbox"/> Moving Update Status - Reply + Mention from scoped to verified	<input checked="" type="checkbox"/> ↓
1.1	<input checked="" type="checkbox"/> Specify Use-Case Module:Update Status - Reply + Mention	<input checked="" type="checkbox"/> ↓
1.2	<input checked="" type="checkbox"/> Realize Use Case:Update Status - Reply + Mention	<input checked="" type="checkbox"/> ↓
1.3	<input checked="" type="checkbox"/> Develop A Component:Update Status - Reply + Mention	<input checked="" type="checkbox"/> ↓
1.4	<input checked="" type="checkbox"/> Execute Test:Update Status - Reply + Mention	<input checked="" type="checkbox"/> ↓
- 2	<input checked="" type="checkbox"/> Moving Follow friend - Basic Path from realized to verified	<input checked="" type="checkbox"/> ↓
2.1	<input checked="" type="checkbox"/> Develop A Component:Follow friend - Basic Path	<input checked="" type="checkbox"/> ↓
2.2	<input checked="" type="checkbox"/> Execute Test:Follow friend - Basic Path	<input checked="" type="checkbox"/> ↓

1.3	<input checked="" type="checkbox"/> Develop A Component:Update Status - Reply + Mention	
	Update Status - Reply + Mention	Implemented
	Some Component	Implemented
	Develop A Component	Test-first development
	Component Description	Required Interfaces Identified
	Source Code	Operations Implemented
1.4	<input checked="" type="checkbox"/> Execute Test:Update Status - Reply + Mention	

Set the Approach

Set required levels of detail

# Apt Automatic Generation of Activities

## Apt Tasks

Dave Cuningham | Quick Search

Dashboards | Projects | Issues | Administration | + Create Issue

Apt for Java version 2.22 is released. This release introduces the Apt Root Cause Analysis Report in the Apt Tasks service, JIRA. For more, consult the Apt for Java 2.22 release information.

### EA-TFS Plugin Requirements - Select and Prioritize use cases - DC

Edit Assign Comment More Actions Start Progress Workflow View

**Details**

Type:	Sub-task	Status:	Reopened
Priority:	Major	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	116, Release 4
Component/s:	None		

Workflow Notice: **Important Notice:** This project's work-flow enforces the following rules:

- Tickets must be Resolved before they can be Closed
- Tickets can no longer be Resolved with outstanding Original Estimates or Remaining Estimates

Thing To Do:

#### Select and Prioritize Use Cases

EssUP-1-5

Project | **Specified System** | Tasking

Project Lead (accountable)  
Customer Representative  
Developer  
Tester  
Analyst

Specified System  
Use-Case Module  
Use-Case Specification

Select and prioritize use cases, use-case flows, supplementary requirements and scenarios to:

- Fit within the time and budget available.
- Deliver the highest value to the users
- Reduce the risk of not reaching a feasible solution in time and within cost
- Demonstrate critical project progress or understanding of needs.

This activity is completed when:

- Use-Case Modules Scoped.
- Use-Case Specifications to the Bulleted Outline state.

This activity contributes to:

- Specified System: Shared.

Recommended approaches:

- Choose scenarios to demonstrate the achievement of project milestones
- Customer prioritizes flows
- Team prioritizes flows

Reload Original Guidance More Activity Guidance

No approach has been selected for this Activity

Things To Produce: The following work products must be completed to the described levels of detail to complete this Activity following the project's configured process (EssUP-1-4). [IssUP-1-4](#)

**Use-case specifications** to the level of detail: **Bulleted Outline**

A use-case specification describes how an actor uses a system to achieve a goal and what the system does for the actor to achieve that goal.

The use case must be outlined in order to understand the overall complexity of the use case, determine its importance to the technical solution and enable effective scope management. At this point the use-case specification is documented as a series of simple steps, typically presented as a bulleted list.

**Completion Criteria**

- The basic flow of the use case is described using simple sentences organized into sequential steps
- The alternative flows are identified and named.

Available Downloads

**People**

Assignee: Dave Cuningham  
Reporter: Francesca Riches  
Participants: Dave Cuningham.

Votes: 0  
Watchers: 0

**Dates**

Due: 26/Oct/12  
Created: 25/Oct/12 4:39 PM  
Updated: 07/Dec/12 1:07 PM  
Start Date: 26/Oct/12  
Since Last Reply: 33 weeks, 4 days ago

**Time Tracking**

Estimated: 30m  
Remaining: 30m  
Logged: Not Specified

- Manage Application Transition
- Application Value Assessment
- Test Management
- Manage Risk
- SOA Service Definition
- SOA Service Architecture Extension
- SOA Service Use Case Extension
- Business Use Case\*
- Fujitsu Architecture Development Method
- Manage Application re-platforming \*\*

# Use for Root Cause Analysis



Standard Root Cause		Where defect Raised														Total for each root cause	Defects located at stage	Stage Gap Total	Average Stage per defect	
Standard Root Cause	Root Cause (example)	Explore Possibilities	Understand the Need	Specify The System	Shape the System	Shape the Service	Shape Support	Implement Software	Build Infrastructure	Establish Support	Test the System	Release The System	Accept the system	Accept the Service	Deliver Service	Deliver Support				
Explore Possibilities	Benefits	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	15	73	4.87
Understand the Need	Contract Obligation		1	1	1	1	1	1	1	1	2	1	1	1	1	1	15			
	Business Requirement		1	1	1	1	1	1	1	1	1	1	1	1	1	1	14	29	119	4.1
Specify The System	Software Requirements		1	1	1	1	1	1	1	1	1	1	1	1	1	1	13			
	Infrastructure Requirements		1	1	1	1	1	1	1	1	1	1	1	1	1	1	13			
	Test Cases		1	1	1	1	1	1	1	1	1	1	1	1	1	1	13			
Shape the system	Service Requirements		1	1	1	1	1	1	1	1	1	1	1	1	1	1	13	52	176	3.38
	Architecture			1				0	0	0	211	0	0	0	0	0	211			
	Design - Platform			1				0	0	0	0	0	79	0	0	0	79			
	Design - Software			1				0	0	0	0	0	0	0	0	0	0			
	Design - Tests			1				0	0	0	0	0	0	0	0	0	0			
Shape the Service	Design Database			1				0	0	0	0	0	0	0	0	0	0	290	738	2.54
	Design - Service					1		1	1	1	1	1	1	1	1	1	9	9	26	2.89
Implement Software	Design Support						1	1	1	1	1	1	1	1	1	1	10	10	31	3.1
	Development - Build Scripts							1			1	1	1	1	1	1	7			
	Development - Code							1			1	1	1	1	1	1	7			
	Design							1			1	1	1	1	1	1	7			
	Data							1			1	1	1	1	1	1	7			
	Test Script							1			1	1	1	1	1	1	7			
	Test Data							1			1	1	1	1	1	1	7			
	Clg Mgt - Config Data Error							1			1	1	1	1	1	1	7			
	Integration - Build							1			1	1	1	1	1	1	7	56	168	3
	Build Infrastructure	Environment							1			1	1	1	1	1	1	7	7	21
Application Baseline									1	1	1	1	1	1	1	1	7			
Application Inventory									1	1	1	1	1	1	1	1	7			
Knowledge Base									1	1	1	1	1	1	1	1	7			
Establish Support	Operational Support Guide								1	1	1	1	1	1	1	1	7			
	Instruction Manual								1	1	1	1	1	1	1	1	7	35	105	3
Test the System	Test Specification										1	1	1	1	1	1	6			
	Test Results										1	1	1	1	1	1	6	12	28	2.33
Release the System	Release Description											1	1	1	1	1	5	5	8	1.6
Accept the System	System Acceptance Checklist														1	1	3	3	3	1
Accept the Service	Service Acceptance Checklist															1	3	3	3	1
Deliver Service	General - Network Change														1	1	3			
	General - Operational Change														1	1	3	6	6	1
	General - Hardware Fault														1	1	3			
Deliver Support	General - User Knowledge														1	1	3			
	General - User														1	1	3			
	General - in Procedure														1	1	3			
	Control														1	1	3			
	Not Analysed		1	1	1	1	1	1	1	1	1	1	1	1	1	1	16	16	89	5.56
	General - Unknown		1	1	1	1	1	1	1	1	1	1	1	1	1	1	16	16	89	5.56
Defects located at stage		2	3	5	9	9	9	11	11	12	236	27	107	28	37	73				
Planned Cost Per Stage		120	567	988	55	88	100	100	100	100	100	100	100	100	100					
Defect correction effort (Hrs)		2.00	2.00	7.00	8.00	20.00	30.00	31.00	44.00	6.00	8.00	4.00	5.00	4.00	10.00	181.00				
Cost Of Rework - (hrs x rate)		###	###	315.70	###	902.00	1,353.00	1,398.10	1,984.40	270.60	360.80	180.40	225.50	180.40	451.00					
[Causal]		###	###	>672.30	###	814.00	1,253.00	1,298.10	1,884.40	170.60	260.80	80.40	125.50	80.40	351.00					
Average cost per defect																14.10				
Defect Density - (proj hrs per defect)																1.73				
Defect Density - (defects per hour)																0.58				
1000																Total project hours				
£45.10																Average project staff rate				
579																Total Defects				

- Enables distributed teams to be highly effective despite being in different countries and time zones
  - Clear and effective way of working for teams to follow
  - Clear link between project requirements/solution complexity and effort/timescale
  - Fast project start-up at lower cost
  - Real-time visibility of status and progress for management
  - Modern, attractive environment for staff and clients
  - Encourages a lean approach to application services
- Robust, reliable development infrastructure
- Integration of methods and tools saves time and cost whilst giving higher quality and productivity

FUJITSU

shaping tomorrow with you

Fin

