
■ Introduction

Broken Process

IF YOUR PROCESS isn't broken, don't fix it! That is to say, if nobody wants to leave your team and it reliably delivers the best possible value to the business in the required time without any unexpected quality or cost issues, think carefully about adopting Visual Studio Team System (VSTS) or becoming an Agile team. This book is not written for people who already have a sound software development process, but rather for those who want to change their process because it is broken in some way.

Welcome to the OSPACS Team

The OSPACS team is entirely fictional and none of its members are intended to represent actual people. However, the team's problems are typical of those we've actually encountered over the past couple of decades while working for numerous organizations, and the characters are woven together from some of the individuals we've met during this time. Therefore, we have no reservations about treating the OSPACS team as though it were real.

Team Background

The OSPACS team is part of a small IT company in the healthcare business. Eighteen months ago it entered into a joint venture agreement with the Old Sainsbury (OS) Hospital for the development of a new Picture Archiving and Communication System¹ (PACS) intended to capture, store, and display digital images such as X-rays and ultrasound scans. The team released

1. Find out more about PACS at the AuntMinnie Web site (www.auntminnie.com).

the first version of this system to the hospital three months ago. However, it was not a great success, as the team had gone three times over its budget, delivered the product six months late, and then spent another two and a half months fixing its bugs.

The team's morale reached a low point a month ago, mainly because everyone was so exhausted after working weekends and late into the night to get the system into production. They then lost two key programmers to a competitor and rumors started circulating that more people were ready to follow them. However, we join the team as its outlook is starting to look a bit brighter, because it has been more than two weeks since the last serious bug was found and the system has finally gone "live" in the hospital's radiology department. Therefore, people are no longer working overtime and can start considering the next phase of the project, which is the development of a generic version of the system for deployment into other hospitals. Mike Hancock (CEO) recently showed his support for this new work by hiring a new senior programmer (Peter Powell) as well as agreeing that the team could upgrade its MSDN subscriptions to get Visual Studio Team Suite.

Current Organizational Structure and Personas

The OSPACS team is organized into a typical hierarchy so that its three programmers report to the project manager, who then reports to the CEO. The business analyst and test manager also report directly to the CEO, which effectively creates three separate departments: programming, test, and product development.

CEO: Mike Hancock



Mike is a serial entrepreneur who set up the company after selling his previous business to a large multinational. He spends most of his time driving the sales force and doesn't really understand why he can't manage the software team in the same way. However, Mike realizes that software people are very different from his salespeople, so he has learned to keep his distance from the OSPACS team.

“The team gives me a project plan that schedules its every task for the next three months, but a week later it’s already out-of-date.”

“These software people cost me a fortune, and I haven’t a clue what they do all day.”

Business Analyst/Sales: Sally Thompson



Sally worked as a business analyst for a major medical equipment manufacturer before she joined the company three years ago. The extensive knowledge she has about the business, combined with her skill at handling customers, means she has a joint business analyst and sales role reporting directly to Mike.

“If 80 percent of the value lies in 20 percent of the features, why do we bother developing 80 percent of our software?”

“Our project manager seems to spend most of the day hiding in his office, producing reports that nobody reads.”

“The opportunities that arise have usually gone by the time we’ve developed the software.”

Test Manager: Maggie Smith



Maggie was one of the original employees of the company and is good friends with its most senior programmer, Sarah Brown. However, this doesn’t stop her from being very critical about the poor quality of the software landing on her desk from the development department.

“Sometimes when writing my tests, I just have to guess what the program is supposed to do. Sally needs to spend time helping me understand the customer requirement.”

“Programmers don’t seem to think that testing is part of their job description. There must be a better way of doing things.”

“There’s a cultural chasm between the development department and the test department.”

Project Manager/Architect: Tom Stanton

Tom is the intellectual powerhouse of the company and knows a lot about different types of software engineering methodologies. However, so far this knowledge hasn't helped him create a successful team.

"People are very dissatisfied with their jobs, and poor tools don't help. I need to fix this problem fast, as we can't afford to lose anymore people."

"Rebuilding all the components takes us days, and then getting them to work together takes weeks of effort."

"I don't know why Sally produces all these long functional specifications, because none of the programmers reads them."

Senior Programmer: Sarah Brown

Sarah is the only programmer left from the original group that started OSPACS. Her software development skills are largely self-taught and she prefers writing code to producing elegant documents and Unified Modeling Language (UML) drawings. Without Sarah, there would be no system and no OSPACS team, so she is highly regarded by everyone, particularly Mike.

"Everyone works on his own separate components because people lock files out of SourceSafe for weeks at a time. The team just doesn't share a common code base."

"We always talk about restructuring the code for the next release, but when the time comes, there's always something else that takes priority."

"Deploying software into our client's environment is always a nightmare because we ship so infrequently that each time it feels like we've never done it before."

"The gap between us releasing each batch of code to testing just keeps on getting longer."

Senior Programmer: Peter Powell

Peter joined the team a couple of weeks ago. Before that, he spent five years working as a contractor for a variety of blue-chip companies. He has already established a reputation for being a bit of a geek who spends most of his time plugged into an iPod, writing code that no one else understands. Peter knows a bit about Extreme Programming (XP), though he has not yet put it into practice in a real project.

“The team needs to feel ownership of the process rather than feeling owned by it.”

“When I look at the existing OSPACS code it’s full of stuff that doesn’t seem to have any real purpose. I’m not sure what’s working code and what’s still under construction.”

“It seems to me that Visual Studio Team System, put into the wrong hands, could stop a project dead in its tracks.”

Junior Programmer: Luke Harrison

Luke joined the company straight after graduating from college, where he earned a computer science degree. Initially he found it very difficult to relate his studies to what was happening in the company, but he has since come to realize that there’s a difference between what happens in textbooks and what happens on a real team.

“Our installation program always works fine on my PC, but it often fails when we run it on our customer’s machines.”

“We spent months creating UML models for the first version of the product and then threw them away a week into coding. Our design work simply doesn’t survive into implementation.”

“We haven’t really got a process—certainly not one that works.”

The Team's Road Map for Fixing Its Process

Tom had first read about XP in the April 2004 edition of *MSDN Magazine*,² but at the time he put off introducing these ideas to the team because he was too busy. However, he now realizes that some drastic action is required, so transitioning the team to Agile development is now at the top of his agenda.

Last week Tom contacted some people from his local Agile community and persuaded them to explain their philosophy to the team during a few lunchtime sessions. The OSPACS team liked what it heard about Agile. However, the team decided to tackle its problems by taking a series of small evolutionary steps, because as well as developing the new generic version of its system, it also has to support the one installed in the Old Sainsbury Hospital. Therefore, following the advice given by one of the Agile experts it had met, the team identified its main problems as shown in Table I-1 and agreed to address these issues by undertaking the corresponding activities over the next six to nine months. In this way, the team hoped to build up its expertise and at the same time reinvent itself as an Agile team.

■ TIP

Contact your nearest Agile user group³ to find people who might help your team make its transition to Agile. These groups usually attract people who are using some form of Agile process in their work and want to share this experience with others.

2. *MSDN Magazine* Web site (<http://msdn.microsoft.com/msdnmag>).

3. The Agile Alliance Web site lists user groups under Resources (www.agilealliance.com).

TABLE I-1: The OSPACS Team's Road Map

Team Building	
<p>"People are very dissatisfied with their jobs, and poor tools don't help."</p> <p>"There's a cultural chasm between the development and the test departments."</p>	Apply Sharp Tools and Values
<p>"We haven't really got a process—certainly not one that works."</p> <p>"The team needs to feel ownership of the process rather than feeling owned by it."</p>	Introduce Agile Development
Achieving a Consistent Level of Quality	
<ul style="list-style-type: none"> "The team just doesn't share a common code base." 	Use Version Control
<ul style="list-style-type: none"> "Rebuilding all the components takes us days, and then getting them to work together takes weeks of effort." "The gap between us releasing each batch of code to testing just keeps on getting longer." 	Build and Integrate Often
<ul style="list-style-type: none"> "Programmers don't seem to think that testing is part of their job description." "Full of code that doesn't seem to have any real purpose." 	Practice Test-Driven Development
<ul style="list-style-type: none"> "Our design work simply doesn't survive into implementation." 	Explore by Modeling
Satisfying Customer Requirements	
<ul style="list-style-type: none"> "When writing my tests I just have to guess what the program is supposed to do." "I don't know why Sally produces all these long functional specifications, because none of the programmers reads them." 	Implement Customer Testing

Continues

TABLE I-1: *Continued*

Releasing Software on Time and on Budget	
<ul style="list-style-type: none"> • “The team gives me a project plan that schedules its every task for the next three months, but a week later it’s already out-of-date.” 	Estimate, Prioritize, and Plan
<ul style="list-style-type: none"> • “We ship so infrequently that each time it feels like we’ve never done it before.” • “Our installation program always works fine on my PC, but it often fails when run on our customer’s machines.” 	Practice for Deployment
Delivering Business Value	
<ul style="list-style-type: none"> • “If 80 percent of the value lies in 20 percent of the features, why do we bother developing 80 percent of our software?” • “The opportunities that arise have usually gone by the time we’ve developed the software.” 	Provide and Reveal Value

■ WARNING

Do not expect any instant results from adopting Agile practices and tools such as Visual Studio Team System. It often takes a year (or even longer) for a team to get into the sweet spot of Agile development, though some of the benefits usually show up much sooner.