**Case #1: Pair Programming Helps But Manager Objects**

You work with four people on a project which is behind schedule and over budget. You were brought in two weeks ago by your manager as a technical leader to help with project completion.

The project goal is to port a legacy system to a new environment. Code quality is low. Explanations about what the system is expected to do are coming in the form of 200 bug reports.

On day 1, you proposed that all debugging, testing and programming be done in pairs. The team agreed and has been enjoying some (modest) results as of today.

On day 3, your manager privately asked you to stop pair programming, as the project schedule is already very late. You tried to convince him that pair programming was good for the team to no avail.

You and the team observed that pair programming gets you better results than solo programming, and would like to do it most of the time.

Last week you had a second discussion with the manager about the very same subject and were not able to change his mind about pair programming.

Yesterday, your manager had one-on-one meetings with every member of the team. Debriefing with the team members today, you learn that he told everyone to stop pair programming.

Every team member wants to continue pairing but is unwilling to go against management.

**Case #2: Scrum Is A Command And Control Method**

John, a young manager, works at a large company where he has delivered projects on time and communicates well with top management.

He was given a team of developers who must work in agile. The agile approach is viewed with interest by the CIO. Marketing, the internal customer, is the main proponent of agile because it produces results quickly and cheaply.

John cares a lot about reporting and he knows that Scrum provides indicators for team performance. As a result, he asks the team members to update their stories and tasks daily so that the burndown chart stays current. John is also very strict about keeping meeting timeboxes.

Lou, the product owner, challenges the team’s story point estimates. He thinks that some of the stories that are sized as large are actually quite easy.

Because of these experiences, the develop team concludes that Scrum is a command and control method.

You arrive in this context to coach the team for “high performance”. John’s objective is to increase visibility and he expects you to support this objective. Lou has little hope…

**Case #3: Scrum Does Not Provide Transparency**

Mark manages a department with several development teams.

One of these teams is an agile team which adopted agile without support from Mark’s department. This teams says they have been doing agile for many iterations and are an ‘advanced’ agile team but nobody outside of the team understands what the team is doing or what the team has completed.

Art is the team’s customer and says he is happy. Or, at least, he is not complaining. However, Art does not know when the software will be completed.

You are responsible for coaching the team at Mark’s request. His objective is to understand what the team is doing. To him, Scrum appears to be a black box approach to software development.

**Case #4: User Stories Are Unclear**

Alan is an IT person who is very enthusiastic about agile. He has decided to adopt agile on all of his projects. His development teams also want to do Scrum. They have had previous experience with agile.

Maria is the Product Owner with a background as a business analyst. Alan was taught that the PO provides specifications in the form of ‘simple’ user stories but he does not know what ‘simple’ means.

Maria has other responsibilities outside of her role as PO. She does not understand how to manage the scope of the application and she cannot give priorities without consulting with business managers. They tell her that everything is a priority.

The development team believes that the user stories are not well written. Maria is frustrated because she is not getting what she wants. The development team responds by saying that she needs to provide acceptance criteria and that the stories are unclear.

**Case #5: Agile Teams Do Not Produce Test KPIs?**

An audit has identified that CRM applications were not properly tested. Because of this audit exception, IT management has made testing a high priority.

Management asked for test KPIs so they can show the Audit Office that IT is making progress. But the agile teams are not providing the KPIs.

Ron, a project manager who doesn’t support the KPI initiative, wants the agile teams to deliver quickly so they meet client requirements.

Furthermore, teams don’t see testing as a valuable activity.

Test are done manually and test execution is slow and inefficient. The lack of test environments creates a bottleneck.

The SQA teams complain that applications are delivered too late in the cycle. The SQA team often has only a week to test the software before it goes into production. Muriel, the head of the QA team, is tired of this situation.

All of the teams are passing the buck. Ron and the development team believe that quality is up to the QA team. Muriel believes that the development team needs to care about quality.

Ron’s manager has to report the requested KPIs. He called you in as an Agile coach to make all of this work together.

**Case #6: The relationship between Business and IT is at a low point.**

For Peter, a Senior VP on the business side, IT costs a fortune and produces little. IT complains that the business does not tell it what to build. The relationship is strained.

It wonders if Scrum can help. But the business has said that it cannot provide Product Owners. The business says, “We cannot

take any action without filling out a software form. IT people don’t start development until specifications are signed off. Afterwards, development takes ages! Agile is just the latest IT buzzword? Agile just means that they want the business to do IT’s job. We have already told them what to do! They don’t need Product Owners! Agile is just like CMMI and ISO 9001 and will soon be forgotten.”

Patrick, an IT Manager who struggles to complete projects, finds the business unbearable: “We do not know what they want, they are never happy and they are completely disorganized. It is impossible to make them follow a process; they are never ready to validate the specifications so IT teams can begin development. Obviously, our relationship is troubled. Could Agile possibly fix things? But how can we tell them that they must be involved in development? They have already said that it does not suit them and they want us to provide the Product Owners.”

You are the Agile Coach called in by Patrick to try Agile as a last chance to get out of this mess.