**SSW 555**

**Agile Methods for Software Development**

**Homework 12**

**The agile practices used at Spotify have been adapted to their company's product, market and staff. That is, they work well for a company that provides a new kind of music service using changing technology.**

**FDD was developed for a different type of project. Could the Spotify collection of practices work as well for that type of project? Compare FDD to the Spotify practices in dealing with:**

**a. requirements collection and use**

**b. project planning and control**

**c. architecture and design creation and maintenance**

**d. implementation**

**e. testing and quality control**

**f. installation and support**

**on a project to develop a financial services system for a large banking firm. In each case assume that the software is being developed by a consulting firm hired by the banking firm.**

**Answer:**

**a. requirements collection and use**

In FDD,

* The domain experts and the Chief Programmers form a modelling team. They study the requirement documents and develop an overall model for the system based on the requirements.
* Further, the overall object model is refined and model notes are written. So there is detailed documentation for this process.

In Spotify,

* Each squad has end-to-end responsibility for the things they do, from design to deployment and beyond.
* The agile coach and the team have full autonomy to decide the things that they are going to design and all the data gathering related to it.
* Autonomy in this context means having the freedom to decide:
  + What to build;
  + How to build it; and
  + How to work together while building it.
* The decisions a squad takes must naturally align with the squad’s long-term mission, their internal goals, overall product strategy and other short-term goals which are reviewed every quarter.

**b. project planning and control**

In FDD,

* After the Chief Programmer builds a Feature List, form the the planning team which consists of the Development manager and the Chief programmers.
* After this they build a development sequence and assign business activities to chief programmers and classes to developers.
* Thus, in FDD there is a specific plan for planning the project and there is a control over all the resources assigned to some of the team members.

In Spotify,

* It is all up to the Squad to plan the activities for their project and there is no such rigid plan as in FDD.
* Here the concept of autonomy is practiced and it is expressed in practice by giving squads the ability to bypass committees and layers of management and make and act upon the decisions the members reach together.
* This allows the company to scale easily without being held back by dependencies and coordination problems.
* There is a place for leaders in their company structure, however he emphasizes that this role is more accurately described by words like servant-leader, coach and mentor, rather than the more traditional boss or manager.

**c. architecture and design creation and maintenance**

In FDD,

* The Chief Programmer designs the architecture of the system. This takes place with the domain experts in the design first phase which is designing the overall model.
* There are regular builds in FDD in which all the integration problems are resolved so the system maintenance is carried out on regular basis.

In Spotify,

* Squads are given freedom to design the architecture of the system they will design.
* Also releases are made a routine so system maintenance carried on a regular basis.
* In keeping with agile principles, Spotify believes that releasing new versions of a product should be done on a regular basis and it encourages squads to be constantly adding new features to its product, no matter how small or incomplete they are.
* The system owner is the “go to” person(s) for any technical or architectural issues related to that system. He is a coordinator and guides people who code in that system to ensure that they don’t stumble over each other.
* He focuses on things like quality, documentation, technical debt, stability, scalability, and release process.
* The System Owner is not a bottleneck or ivory tower architect. He does not personally have to make all decisions, or write all code, or do all releases. He is typically a squad member or chapter lead who has other day-to‐day responsibilities in addition to the system ownership. However, from time to time he will take a “system owner day” and do housekeeping work on that system.

**d. implementation**

In FDD,

* The feature teams led by the Chief programmer are responsible for implementing the features.
* Each team contains all the owners it needs to create the feature.
* Owners may be on multiple feature teams at the same time.
* Chief programmers may also be class owners.

In Spotify,

* Each squad has end-to-end responsibility for the things they do, from design to deployment and beyond.

**e. testing and quality control**

In FDD,

* FDD inspections usually performed by feature teams on their own features.
* During this process the system designs and code are inspected.
* Code inspections are carried out on regular basis.
* Configuration management - in which code is maintained to manage conflicts and maintain consistency.

In Spotify,

* Squads are encouraged to apply Lean Startup principles such as MVP (minimum viable product) and validated learning. MVP means releasing early and often, and validated learning means using metrics and A/B testing to find out what really works and what doesn’t. This is summarized in the slogan “Think it, build it, ship it, tweak it”.
* Squads carry out Sprint retrospective meetings on a regular basis in which the teams discuss on what worked well and what went wrong.
* Spotify encourages finding ways to ensure fast failure recovery rather than simply avoid failing at all.
* Some squads even use a fail wall where they publicly share their failures and whenever there is a failure, the squad conducts post-mortems and retrospectives which allows it to capture any learnings from the experience.

**f. installation and support**

In FDD,

* In FDD the domain experts are the voice of customers who convey to the feature teams the problems that the clients face with handling the systems.

In Spotify,

* The system owner is the “go to” person(s) for any technical or architectural issues related to that system.
* He focuses on things like quality, documentation, technical debt, stability, scalability, and release process.
* Thus, the installation and support systems are handled by the system owner.

Thus, after comparing the above aspects of FDD and Spotify we come to a conclusion that,

**the consulting firm should keep continuing using FDD rather than Spotify process because:**

* The requirements are stricter in banking domain and there is very little scope for going out-of-the box or creative additions.
* Also, banks have compliance requirements and this being a consultant firm everyone in the team may not have access to all the information. As a result there has to be a Chief Programmer or Development Lead from the bank who guides and directs all the feature teams.
* Also there should be a concrete management hierarchy as in FDD to manage all the resources.
* As a consulting firm, they have to finish the task in stipulated amount of time and have stricter deadlines to follow.
* Some aspects from Spotify method could be adopted like cross pollination, sprint retrospective meetings.
* Feature teams assigned to work on each aspect of the banking domain could be much clarity to the functioning of the project rather than squads in Spotify.