PROJECT MANAGEMENT ESSAY

MATTHEW DYSON

Project management patterns form a source of valuable knowledge in which past experience is broken down into a series of situations and solutions, which can be easily searched and linked together to form a network of ideas through which managers can evaluate and assess the suitability of common approaches for implementation in their own situations. Once implemented, managers can contribute their own experiences back into the pattern database, which helps to expand and increase the amount of real-world information that the patterns contain, and through repetition of this process the quality and quantity of patterns will dramatically increase. There also exists a wide variety of defined and commonly implemented project management methods which are given as a strict structure that the manager will mould their team to, with different methods tailored towards different requirements. For instance, some methods will assume fixed requirements and flexible timescales, whereas others will assume a fixed time for completion allowing for flexible requirements. Once one of these methods is implemented, a strict set of decision making criteria is implemented that ensures that the project will not make compromises in the wrong area.

When overseeing a project, a manager should be focusing on two key areas – the social interaction of their team, and the progress that they are making on the tasks given, which ideally should not be mutually exclusive, as outlined by Bales in (1). Within this essay, I will be examining two methods of managers providing solutions to each of these areas – utilizing patterns, and using more traditional methods such as the Dynamic Systems Development Methods (DSDM).

Within a project team, it is extremely important to ensure that members are co-operating on a personal level before trusting them with a business-critical assignment. Weingart & De Dreu (2) suggest that conflict will "interfere with team performance and reduce satisfaction… and distract team members from performing the task", which obviously provides a significant risk to be considered by a project manager before starting any project. For instance, having a team member under pressure from their peers during project development could lead to that member not putting their full effort into their work, as they are feeling oppressed and unable to perform for fear of reprisals. A strategy for combating this is suggested in pattern 2008-22 – "Overpowering Team Member", which calls for quieter team members to be "addressed directly to 'force them' to contribute". Whilst this blunt approach may be effective in some instances, from personal experience I believe that a better approach is to address personal relations through individual contact with members, as "forcing them" to contribute may introduce resentment towards the manager, making the situation harder to resolve. Close personal relationships between the manager and the team members can be extremely useful in order to locate and remove any issues before they cause damage to the teams' progress. Kelly and Barsade also argue the usefulness of working with individuals in (3), where they present the case for working from the 'bottom-up' with a group, as individual moods and emotions of members has a massive impact on the groups' performance.

This idea of concentrating on the individual before the overall group is also a key part of the SCRUM (4) method of project management, where regular meetings are called to address any issue that the team member may be having. These meetings may not be the most relevant place to address any personal and social issues within the team, however by adapting this process to include a one-on-one meeting with the Scrum Master and the individual team members where such issues could be raised. Whilst SCRUM and some other defined project management methods include this regular personal contact with team members, the actual strategy for dealing with social issues is not defined. Because of this, any manager faced with a social problem within their team needs to be able to draw on past experiences, whether this is personal or from a source, which is provided by the project management patterns listing. This does, however, depend on the project managers' receptiveness to external knowledge; Douglas Adams wrote that "human beings, who are almost unique in having the ability to learn from the experience of others, are also remarkable for their apparent disinclination to do so" (5). From my own experience, being able to find similar situations to the one I'm presented with from the pattern listing is extremely reassuring, however blindly following the recommendations can cause more problems than they solve. For instance, when initially forming the team, the pattern 2007-1 "Headless Chickens" suggests assessing individual members' prior skills before facilitating discussions about who may be most suited to each role necessary, ultimately allowing the team to make a democratic decision. However, I believed this would lead to bitterness within the team, as they were beginning to form two factions with two more vocal members taking the lead in each. I ultimately chose to combine the suggested approach from 2007-1 with 2008-11 "Sub-divided Teams" in order to use this fracture in the teams' social structure to the advantage of each project (SEG & Phoenix), as the skills held on each side were split near-optimally for the different objectives that had to be met. Although this did not lead to a clearly defined structure within the team, there is still enough of a hierarchy to avoid confusion, and has avoided the two factions being driven further apart. This approach clearly demonstrates the need for patterns to be taken with 'a pinch of salt', and that combining and adapting them to individual situations is more useful than taking them as read.

No two engineering projects, by design, are ever the same; therefore the usefulness of patterns as information sharing between projects is highly debatable. There is evidence suggesting that common ground can be achieved between projects within specified fields – such as the paper 'Patterns as a Means for Intelligent Software Engineering' by Deugo et al. (6), which gives a number of situations in which patterns can be applied in different implementations within the software engineering discipline. However, it is extremely difficult to extrapolate technical issues across different areas of expertise – for instance the patterns suggested by Deugo et al. would be completely irrelevant in an architectural project, and vice versa. It is therefore reasonable to assume that technical approaches and problems are better targeted by traditional project management methods; however there are still some exceptions to this. For instance, the pattern R7 "Loss of work" gives details on how to avoid the risk of data loss by implementing pre-emptive backup solutions through the use of configuration management and multiple levels of redundancy. As this pattern takes a more general view of the problem, rather than keeping it within a specific context, it is more useful when attempting to utilize patterns over a wide range of fields and specialisms, hence being appropriate in describing technical approaches to a project as it lays an overall framework that can be used in different areas. A contrast to this is the pattern K7 "Technological Failures", which instead gives the specific instance of email problems, and suggests a direct approach to overcome the issue. Whilst this would be extremely useful in the event of the context being met exactly, a more useful approach in the writing of the pattern may have been to take a broader view at technological failures, and suggest a more global risk abatement strategy that can be applied to different fields and situations. This could then be complimented by further patterns, building upon the original ideas, which would drill down further into the specific instance if appropriate. By doing this, patterns may be more useful when defining technical approaches to a project, as past experience is the best way to shape new approaches to problems.

A Guide to the Project Management Body of Knowledge (7) suggests that there are 5 key elements within a project – its initiation, planning, execution, monitoring and closure. These phases are picked up by most project management methods and are implemented in different manners, but keeping the same method, as each phase has its own benefits. For instance: the

Dynamic Systems Development Model describes three distinct phases – "pre-project", "project life-cycle" and "post-project"; the Scrum development model calls for the project being split, each sprint being planned individually, monitored, and then reviewed retrospectively. From this, we can assume that the 5 step process suggested by (7) is an appropriate and tested method of planning a projects progression. I firmly believe that following this process leads to a better understanding from the project team when they are involved in the relevant steps, which in turn leads to a more productive working team, as they are able to see the 'bigger picture' rather than being focussed on their individual work. Creating this element of peer review on the overall project I believe also leads to team members being more personally attached to their work – as an individual working on a single element may not see how relevant their work is to the overall project, whereas if they are aware where their piece fits the puzzle, they are more willing to work on it. This approach of involving the team in the planning stage also helps to combat any potential interactivity problems that may arise when the project is being worked on in parts, as in my experience people who are aware of the overall design, and also of who is responsible for each section of the project, are more likely to collaborate on the actual implementation, hence allowing flexibility on the design, and a more fluid development process which should lead to less problems further down the project timeline. This is re-enforced by the pattern H12 "Code won't integrate", which provides a solution for code failing to meet specifications and failing to tie together – this would be much easier to implement as a preventative measure through project planning as a risk-annulling strategy, rather than as a repair to the project. This leads towards the conclusion that a clearly defined approach to a technical project is better than implementing patterns ad-hoc when necessary, as preventing the risk from occurring is better than fixing in retrospect.

The initial design of a projects lifecycle is not an easy decision to make. Some managers and projects favour an agile approach, whereas others lean towards a more defined strategy such as the waterfall model. The choice between the various styles of structuring the project is paramount to achieving the best result with the requirements, restraints and resources available. For this reason, and through my own personal experience, I would draw a cost/benefit analysis for each appropriate strategy in order to assess the best approach. This bias with prior knowledge in mind is something that favours the use of patterns, which essentially are individual pieces of information based upon someone's experience. However – the exclusive use of patterns to decide upon a project lifecycle is not something that should be undertaken, since each individual project has a number of factors that makes it unique (as stated previously), and hence should be examined against different management structures. This indicates a clear lean towards the necessity of traditional and clearly-defined project management methods when looking at technical approaches to a project.

Within this essay, I have examined the various merits of pattern-based and more traditional methods to project management, which has led towards several conclusions. Overall, the implementation of patterns within any project is useful, as the prior knowledge contained within them is invaluable; however their usefulness is vastly affected by the scope in which they are written. In cases of social issues, having a bank of specific instances and potential solutions from which to draw on can be extremely useful, but this is not the case when looking towards a technical issue, as invariably the technological specifics of an individual project are unique, hence rendering specific patterns near to useless. However, patterns that are written with a more global scope can be tailored to individual projects, and are arguably of no less use for social issues in this instance as well. It is also worth considering that the continued development of patterns is extremely necessary, as without a feedback loop from managers currently implementing and trying the patterns, the dataset will stagnate and eventually become less useful. Managers using the pattern listings should also be attempting to tweak and tailor the methods described to their own needs, and putting the results back into the database, as a pattern implemented without any thought has the potential to be more damaging than helpful, highlighting the need for a manager to be in-tune with all aspects of their teams development (including both social and project progress).

The implementation of specified project management methodologies can also be extremely useful, however more so when viewing the technical aspects of the development cycle, due to the vast range of social issues that can occur, and that most management methodologies do not have any specified concern for the interaction of a team, treating them as individual resources rather than a team. Nonetheless, management patterns can be implemented as a part of these practices, and the two concepts together form a strong project management strategy. Since there are such a wide variety of project management methods, and as they are near-impossible to overlap and implement multiple methods at once, the choosing of a method needs to be a good fit with the existing structure of the team and project, otherwise the method will inevitably fail and cause significant irreparable damage to the project. Guidance in choosing these methods may well come from patterns, however the technical approaches to the project need to be based on a complete tried and tested model rather than applying patterns as and when they are deemed to be necessary.

BIBLIOGRAPHY

1. *Interaction process analysis: A method for the study of small groups.* **Bales, R. F.** Reading : Addison-Wesley, 1950.

2. *Task Versus Relationship Conflict, Team Performance and Team Member Satisfaction: A Meta-Analysis.* **De Dreu, Carsten K. W. and Weingart, Laurie R.** 4, s.l. : Journal of Applied Psychology, 2003, Vol. 88.

3. **Kelly, Janice R. and Barsade, Sigal G.** Mood and Emotions in Small Groups and Work Teams. *Organizational Behavior and Human Decision Processes.* 2001, Vol. 86, 1.

4. **Rising, Linda and Janoff, Norman S.** The Scrum Software Development Process for Small Teams. *IEEE Software.* 4, 2000, Vol. 17.

5. **Adams, Douglas.** *Last Chance to See.* 1991.

6. *Patterns as a Means for Intelligent Software Engineering.* **Deugo, D., et al., et al.** Ottawa : Proc. of the Int. Conf. of Artificial Intelligence, 1999.

7. **Ausgabe, Vuerte.** *A Guide to the Project Management Body of Knowledge.* 2010.