

Team-based projects are an important part of software engineering courses at universities. It is a good experience for students to learn to work together in a team as they will most likely be required to do so when they enter the workforce. Employers of Computer Science and Software Engineering graduates value effective teamwork. “Project experience and teamwork have been identified among the most important knowledge deficiencies of recent graduates, i.e. the most important skills that the prospective employers found were lacking in recent graduates of Computer Science and Software Engineering programs.” (Tafliovich, Petersen & Campbell 2015, p. 494).

Teamwork can be challenging. “Teamwork can mean the difference between success and failure of a software project.” (Marques & Ochoa 2014, p. 99). Many projects are doomed to failure. Some projects are not completed to the level of quality desired by all team members. This essay will seek to identify a number of possible signs that a team-based project may be at risk and look at how team members can identify the presence of risks in their project and how they can recover from them and still achieve success.

Many teams fail to reach their potential. From my own experience I have learnt that things do not always go according to plan. Success can be elusive not just for teams at universities. In the ‘real world’ even well managed projects rarely run exactly to plan (Barker & Cole 2012). There are a number of reasons for this. Theory is different to actual application; management, planning and communication which

are very important to the success of a project are sometimes lacking. In general, team members are lacking in knowledge on how to collaborate successfully (Frankl, Bitter & Kaufmann 2014). Some teams discuss a lot but do not make decisions or are not clear about what it is they are going to deliver. There can be a lack of attention; team members do not always actively listen during meetings and the ability to listen is a skill that affects the quality of communications (Portny 2013). Teams are heavily dependent on each team member delivering their work on time and without errors; team members do not always keep to their timeline and can make mistakes, resulting in them falling behind on delivery targets and goals. There are three main variables affecting team work: “After thirteen years teaching students to develop software in teams (and almost a hundred projects), we have empirically observed that *communication*, *coordination* and *motivation* (usually manifested through personal commitment) are the main variables affecting teamwork.” (Marques & Ochoa 2014, p. 99).

Teams need to set goals and keep to them, and include risk management in their planning which is vitally important to a project. All team members need to actively participate, provide ideas and think about the project as a whole and not just the individual part that they are tasked with. It is not a time to be passive; everyone has to stay motivated and communicate well, discuss issues without conflict and respect each other. “Effective communication among members of a software development team is considered to be a critical factor in

the success of software projects.” (MacKellar 2012, p. 134). Team members will have their own weaknesses and strengths so it is good to find these out at the beginning of the project. Everyone needs to be clear on what each team member is responsible for and never lose sight of the client. There needs to be trust, and the project leader has to lead well and make sure everyone stays on target with their part of the project. A team needs to work well together because “software engineering is a highly collaborative process.” (Frankl, Bitter & Kaufmann 2014, p. 154).

In my own team-based project the communication could have been handled better. In hindsight we should have had more meetings, discussed exactly what everyone should have been doing and made sure that we were keeping to the timeline. The whole team should have interfaced with the client so that everyone could have been better informed of exactly what was required and how to deliver it. Relying on two team members to convey this information accurately let the rest of us down. I should have made an effort to check what the project leader and the rest of the team were doing and if everything was on target for the deadline. We should have all been fully engaged, contributed fully, kept focused, met commitments and divided up the workload equally.

The signs that a project may be at risk are when there is a lack of team focus on the timeline and team members do not keep to plan and fall

behind schedule with their allocated tasks. There may be missed deadlines and no recovery plan. Sometimes there is a lack of good project leadership, different opinions on how to do things and lack of agreement about decisions. Team members may also be lacking morale and there may be tension in meetings. Poor attendance at team meetings is a sign that all is not well (Kerzner 2014). Team members may not report on their progress and technical problems may be slow to identify. The client's requirements may also be misinterpreted, leading to the development of software which is not up to standard and has to be rewritten.

Unfortunately, in my team-based project our project leader did not ensure that all of us were keeping to the schedule. If any of the team members did not deliver their work on time, the project leader should have held those team members accountable to the team. We also lacked the proper hardware for developing the project and the team should have sorted this out at the beginning. No-one enforced the use of version control which would have helped track any potential issues. Instead of being completely focused on my own part of the project, I should have checked to see if everyone else was finishing their part on time and to the required standard.

Other signs that a project may be at risk are when there is a lack of priority, poor planning and when schedules are too optimistic. In my team, our schedule was difficult to keep to for some team members

and therefore we should have planned better and allowed for a contingency buffer in our timeline. I would have liked more details on the design as I was not in touch with the client and communication in the team was poor. Some of my team members lacked motivation. Motivation in a team is very important. In the ‘real world’, team motivation is positively correlated with project outcome and higher motivation increases the likelihood of success: “Australian software engineers are more highly motivated if they have a good project manager.” (Verner, Babar, Cerpa, Hall & Beecham 2014, p.126). Unfortunately, my team did not have a good project manager who motivated everyone. We failed to identify any non-technical risks which resulted in them being left unmanaged.

To identify the presence of risks teams need to include risk and issue management in their planning, which is an essential part of software development. Everyone on the team needs to brainstorm and look at what events, should they occur, would threaten the delivery of the project and plan for any situation that if left would have the same effect. They need to prioritise risks and issues and tackle higher risks first by monitoring and controlling them. Team members should take preventative action and establish what could be done to prevent the risk from happening. Reviewing the underlying assumptions of a task can help identify risks that could be overlooked if its definition is inaccurate or incomplete. There are a variety of ways to incorporate risk management into a project: “risk management theories, risk

games, workshops, risk analysis tools or educational studies.”
(Koolmanojwong 2014, p. 201).

At the University of Southern California, for example, student teams use the Distributed Assessment of Risks Tool (DART) to identify, analyse, mitigate and manage risks. Risks are compiled and analysed to find out their weak and strong points. Team members look at the probability of risk occurrence on a scale of 0–10 and the size of the loss on a scale of 0–10 and use these to calculate the risk exposure by multiplying the probability of loss to the size of the loss. The teams then discuss their mitigation plans with their client. The team leader makes a list of the top ten risks and puts them in a weekly progress report. The teams also use surveys and milestone reports to identify risks (Koolmanojwong & Boehm 2013).

To recover a project once it starts to go off the rails, teams need to reassess everything in a new light and utilise the experience of their members to provide ideas on how to fix it. They need to redefine the project, audit it, find the flaws and come up with a new action plan. Teams need to regain control by planning positive action. Everyone on the team needs to be well-informed, share knowledge, be flexible, commit to solve the problems and technical issues as they arise and keep the project completion within the scheduled timeline. Improved planning and time management is required, prioritising what needs to be done first to keep everyone on track. Teams need to determine

what has to be done first, what can wait and what is not needed. They must have a clear understanding of the steps that need to be taken to stabilise the project. Team morale needs to be rebuilt (Kerzner 2014). It requires everyone to function as a group as well as individually: “High-level group performance can only be achieved through a systemic perspective: the group as well as the individual (being fundamental elements of groups) have to be considered.” (Frankl, Bitter & Kaufmann 2014, p. 154). The earlier problems are found and corrected, the greater the likelihood of a project being recovered.

In hindsight, my team should have replaced our project leader as he failed to keep everyone on track. He let things slide and did not communicate well with the team. The project leader is vitally important in the recovery of a project, not just in university projects but in companies as well: “Almost all organizations surveyed (92%) rated the skill and knowledge of the project manager very important (64%) or important (28%) to the success of the recovery.” (Project Management Solutions 2011, p. 7).

Other methods to help address problems with teamwork include ‘ThinkLets’, which are activities that produce a known pattern of collaboration among people working together towards a goal and recommended practices: *thinking, collaboration, releasing, planning and development practices* (Marques & Ochoa 2014).

In my team-based project, we should have improved communication amongst ourselves with regular updates on how we were progressing, met regularly (even daily near the final week), been more motivated, resolved our unrealistic schedule and thoroughly examined the risks. By completing our work to a strict timeline, the quality of our project would have drastically improved. When it came time to submit our finished project, our team leader should have notified us as to exactly what he had submitted and to whom in order to ensure that our work was being fairly represented and had been delivered correctly, but he failed to do so and therefore his mistake of neglecting the client went unnoticed for a whole week.

When teams use their strengths and work together well, there is no doubt that they can achieve success. Learning good teamwork is a valuable experience for university students to obtain skills to prepare them for a work environment. In my case, I have learnt that team-based projects require extra effort and focus to ensure that everything stays on track. Teams need to have a capable team leader, solid planning and communication and the ability to recover the project if it starts to go astray. Also, if the warning signs are discovered early enough the project can be recovered more easily. The next time I take part in a group project, I will be able to apply the experience that I have gained to ensure a successful outcome.

References

Tafliovich, A, Petersen, A & Campbell, J 2015, 'On the Evaluation of Student Team Software Development Projects', *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pp. 494-499.

Marques, M & Ochoa, SF 2014, 'Improving Teamwork in Students Software Projects', *IEEE 27th Conference on Software Engineering Education and Training*, pp. 99-108.

Barker, S & Cole, R 2012, *Brilliant Project Management*, 3rd edn. Pearson, Harlow.

Frankl, G, Bitter, S & Kaufmann, B 2014 'Learning and Working Together as Prerequisites for the Development of High-Quality Software', *IEEE 27th Conference on Software Engineering Education and Training*, pp. 154-157.

Portny, S 2013, *Project Management for Dummies*, 4th edn. John Wiley & Sons, Inc., Hoboken.

MacKellar, B 2012, 'A Case Study Of Group Communication Patterns In a Large Project Software Engineering Course', *IEEE 25th Conference on Software Engineering Education and Training*, pp. 134-138.

Verner, JM, Babar, MA, Cerpa, N, Hall, T, Beecham, S 2014, 'Factors that motivate software engineering teams: A four country empirical study', *The Journal of Systems and Software*, vol. 92, pp. 115-127.

Koolmanojwong, S 2014, 'Top-10 Risks in Real-Client Software Engineering Class Projects', *IEEE 27th Conference on Software Engineering Education and Training*, pp. 201-202.

Koolmanojwong, S & Boehm, B 2013, 'A Look at Software Engineering Risks in a Team Project Course', *IEEE 26th Conference on Software Engineering Education and Training*, pp. 21-30.

Kerzner, H 2014, *Project Recovery: Case Studies and Techniques for Overcoming Project Failure*, John Wiley & Sons, Inc., Hoboken.

PM Solutions 2011, *Strategies for Project Recovery*. Available from: <http://www.pmsolutions.com/audio/Strategies_for_Project_Recovery_Research_Report.pdf>. [18th June 2015].