exalate

The Project Management Blueprint for Agile Collaboration Between Teams

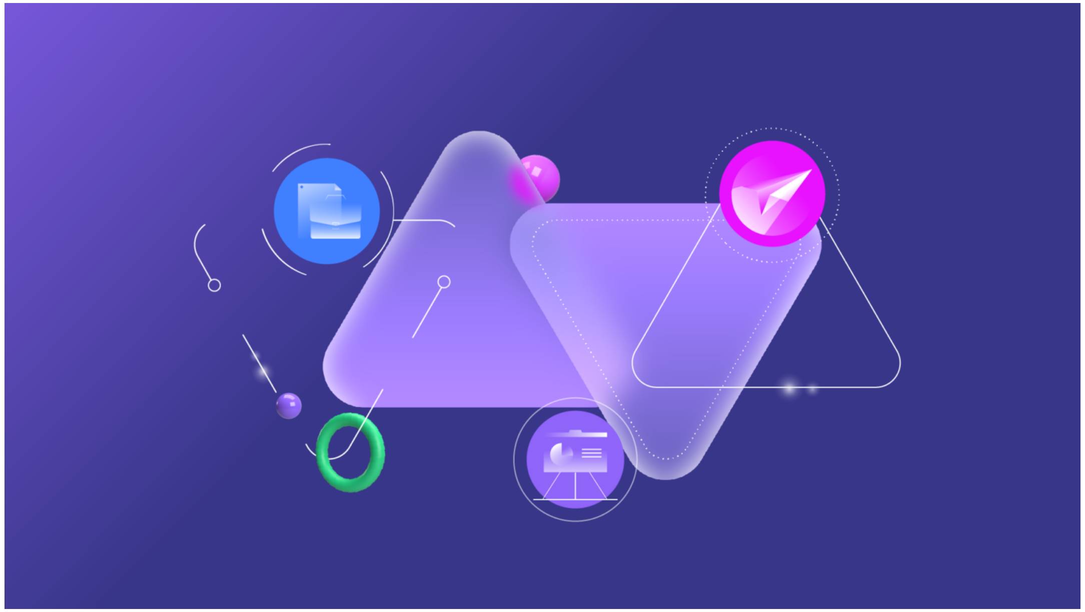

Table of contents

Project management skills in the evolving software development space have an increasingly demanding role. Project managers need to manage their everyday responsibilities as well as achieve the following;

- Supporting a rapid pace of project and product delivery
- Gaining an all-encompassing view of project progress
- Keeping the focus on value while maintaining (or increasing) speed of product delivery
- Managing stakeholder expectations

These may not be new challenges to many Project Managers. But, many are seeking to implement smarter ways of working in order to maintain productivity, collaboration, speed and quality.

Establishing better processes in development projects allows Project Managers to focus on the things that matter: customers and their expectations.

Project Managers are aware that competition is tough, and failure to keep the customer as the focus may lead to losing them altogether.

This whitepaper aims to guide Project Managers through their current challenges and outline realistic approaches to overcoming them. It will offer a set of steps to take, to adopt agile project management methodologies.

Who is this white paper for?

1. Multiple teams working together from multiple locations ...

Whilst the concept of more agile collaboration could be applied to a multitude of disciplines and project management scenarios, the key focus for this paper is on the need for agile working across multiple software development teams.

This is relevant to project managers who are responsible for the coordination of multitudinous and complex factors in the delivery of product development.

These project managers are responsible for coordinating teams of people from both within and outside the organisation.

They are working together towards common objectives but often without one unified approach.

2. Teams using disconnected technology ...

Another common issue for project managers to address is the supporting technology.

In some cases there may be project management tools and applications in place to support and assist teams in collaborating and communicating.

But there is often a lack of consistency. And each team finds themselves using whichever tools they see best fitting their own requirements.

In some instances where subcontractors or small teams are part of the project there may be a complete lack of supporting technology altogether.

3. Project manager with an ever growing team ...

Project Managers in software development often find themselves looking to grow and expand their teams to include other internal teams, external teams and sub-contractors in order to meet the demands of a growing business.

4. Teams using a mix of approaches to managing workloads

In bringing teams together to work on the same project, Project Managers are likely to be faced with a combination of approaches and processes already in place.

One team may be comfortable using the waterfall model to base their workload on. While another team may still be relying on spreadsheets and written task lists.

The Challenges of working in a multi-team environment working from different locations

This often complex scenario which software development project managers find themselves in, produces a host of challenges which need to be addressed;

1. Volume of information

Managing multiple teams, team members, tasks, deadlines and statuses leads to an overwhelming amount of data to coordinate into a logical and simple plan.

In projects with no collaboration and unified approach this can lead to huge risks and delays, in delivering product releases on time and in budget. Especially when the project is being worked on by multiple teams.

2. Inflexibility

The software development world is competitive and fast-paced.

Successful organisations rely on their project teams to deliver regular product releases which respond to changing feature prioritisation and insight from the dynamic software market.

Internal and external influences can affect the scope of projects enormously.

Many development teams find themselves not agile enough to adapt quickly and effectively.

3. Coordination of disparate teams

Managing the individuals within one team is a challenge in itself.

Add to this multiple teams in multiple locations and the challenge becomes overwhelming.

When aiming to deliver successful projects involving a number of internal and external teams, project managers need to consider individual ways of working, differing quality standards, internal conflict, cultural and time differences.

4. Ineffective communication

When multiple teams and individuals are working towards one single outcome, good communication is essential.

And project managers can't rely on ad hoc emails and phone calls to keep their teams on track.

It is the role of the project manager to break down communication barriers and encourage collaboration over working in silo, all the while avoiding confusion.

This can be a tricky balance to achieve.

5. Managing risk

Risk is an inevitable part of any software development project.

Even more so when multiple teams are involved.

Project managers need to be aware of potential risk, manage appropriately, communicate the risk outwardly and mitigate where possible.

6. Maintaining speed of delivery

Software organisations rely on their speed of product delivery in order to survive.

But when each software project encompasses a multitude of moving parts this can be a big challenge.

Project Managers need to ensure there is a strategy in place for maintaining speed and quality (with limited risk) when dealing with patches, updates and changing prioritizations.

The Approach

With the potential pitfalls of disjointed project management in software development displayed clearly, it makes sense that project managers are seeking smarter ways to

- coordinate their teams for better collaboration,
- deliver faster product development with minimised risk,
- increase flexibility to respond to change.

This is where agile project management comes into play.

The **agile methodology** is a broad term used to describe various methods to support an iterative and incremental approach to effective software development.

The goal of the agile methodology is fast, regular and high quality software development facilitated through planning, tracking, analysing, and above all, communication.

This approach to software development is a well-known concept which has been in practice for decades.

One of the most common approaches adopted to support this movement is the Scrum agile development methodology. First proposed as a development framework in the 1980's, Scrum is now used widely in organisations all over the world to improve business (not just development) processes.

Essentially agile development, in the context of software, centers on iterative and incremental development of software products.

There are many methodologies based on the agile concept and **Scrum** is one which is widely adopted throughout software teams worldwide.

The **Scrum-based methodology** focuses on breaking down large projects into allocated tasks and regular sprints (or reviews).

This allows a development team to track progress and add, remove or change elements along the way.

It acknowledges the importance of customer feedback, the need to be flexible, and the need for effective team collaboration.

The importance of scale

What often leads Project Managers to consider ways of improving their processes is the need to scale.

Growing teams lead to new group dynamics and the need to adapt their way of working accordingly.

In this common scenario, it is the role of the project manager to maintain visibility at all times and communicate the goals and objectives clearly.

This is not easy with multiple disparate teams to consider.

For development teams to be effective it is important that they understand how their contribution to the project affects the overall outcome and meets the objectives for the organization as a whole.

Product Architecture

The capability to scale depends on the concise breakdown of a product into components and subsystems or features.

This demarcation allows to distribute the responsibility of the product amongst the different teams.

For instance a product, such as a mobile app, can be broken down into components such as the front end, back end, database reporting function and design.

The Project Manager can gain visibility over these components and their development status, and manage deadlines and priorities more effectively.

System Integration Team

When the product is broken down into components where multiple people (teams) contribute to them, you need to be able to assemble them back together.

The system integration team is responsible to assemble the delivered components into one product.

Successful product development and release relies upon a multi-skilled team to validate every element and orchestrate every release.

Their responsibility includes taking care of the infrastructure for reviewing and testing integrations. And carrying out continuous improvement checks.

One of their main activities is to establish test cases, and define release specific test plans.

Definition of Done

Most project managers will be familiar with the term 'Definition of Done'.

They will also agree that it is important for all team members to understand what the criteria is against which a product is reviewed for release.

Each story or feature in a sprint has its own level of acceptance criteria, and each sprint has multiple stories.

It is therefore important that the project manager provides a way for the team members working on different components, to access a unified set of acceptance criteria for the entire project.

This is essential for a release to go ahead.

The Scrum agile approach supports this unified thinking brilliantly, with its focus on collaboration and incremental progress.

Regular Scrum meetings provide the perfect opportunity to check in on progress against criteria. Enabling the product owner to make a decision about what goes into a release and what ist pushed to the backlog.

The Scrum approach also enables the Project Manager to maintain a level of fluidity to the Definition of Done. Because the open communication channel means that as teams get more familiar with working collaboratively and pick up pace, the Definition of Done can adapt accordingly.

Promotion Levels / Product readiness

With any software product release, it is important that an informed decision can be made if the build can be published.

Promotion levels allow to associate a quality level to a particular build and reflects the type and amount of tests that have been carried out (successfully).

For instance a:

- **bronze** promotion level indicates that all unit and component tests succeeded.
- Performance tests will be carried out to achieve a **silver** level,
- while a **gold** level requires that the documentation matches the provided functionalities.

There will be scenarios where a gold release is imperative. And other occasions when a patch needs to get to market and a bronze build is sufficient.

These promotion levels are key. But decisions cannot be made without full visibility over what the acceptance criteria are. And what needs to be done in order to achieve them.

Supporting adaptiveness and acknowledging the importance of the customer

It is important to stress the point of how an agile framework aligns to the need for adaptable and responsive product development.

An incremental development framework supports short releases through regular sprints and scrum reviews. Giving the development team the chance to review progress and rearrange priorities if necessary.

In addition, essential customer feedback can be reviewed as part of this progress and potentially integrated into a release immediately.

This focus on the customer is an essential consideration for Project Managers.

Achieving agile collaboration

Once a decision has been made to implement an agile approach for the collaboration of multiple teams, what steps can you take to get started?

1. Gain support before you start

The first step to implementing agile collaboration in software development is to ensure that there is **internal support from relevant parties**.

It is very challenging to implement a new framework within an existing organization. And Project Managers establishing collaboration through cross-functional and dispersed teams can often meet resistance. Resistance to change and cultural barriers.

Implementing a Scrum approach is asking the teams to change their focus and take control of their workload in a way which they may not be familiar with.

Support from key stakeholders within the organization displays confidence and assurance. Which can help overcome some of this resistance to change.

2. Establish clear goals

As a Project Manager you may be convinced that an agile collaboration approach is the way forward for your software development teams.

Take the opportunity to think about how Scrum will initially affect the teams and their productivity.

Then establish what you are trying to achieve through the implementation.

List out the goals and objectives for implementing a Scrum and agile approach. And it will be easier to break these down into actionable tasks and responsibilities.

It will also give you a good idea of whether you have all of the right people you need in the teams in order to achieve these goals.

3. Predict challenges and plan how to overcome them

Once you have established your goals and objectives, it is a good time to step back and think about what challenges may arise pre, during and post implementation of this new collaborative approach.

These may be linked to organizational culture, new starters or leavers, or technology problems.

Whatever they are, the more aware you are, the more chance you have to successfully plan ahead to overcome them.

4. Decide on roles

You have clear goals laid out and have predicted the issues you may come across as part of the process.

Now you need to decide how the Scrum roles will be allocated across the members of the teams who will make up the development project team.

This can be a complex task.

While the majority will assume the role of team member, you need to make sure there are clear Product Owners working on the development.

And you will have to decide whether you, as Project Manager, will take on the role of Scrum Master. Or whether this will be someone else that you work in unison with.

Often the role of the Scrum Master changes for each sprint.

Another important consideration is the size and scale of a project. As many large projects require a Project Management Officer (PMO) to manage multiple Project Managers.

Additionally, for collaboration across multiple teams you need to decide whether there is a requirement for nominating Scrum coordinators to take part in the 'Scrum of Scrums'.

When scaling up means that Scrum teams are getting too big, teams can nominate one person to deliver their input and feedback to other representatives in a smaller group.

5. Be transparent at all times

Once you have decided on the Scrum approach and are busy defining goals and allocating responsibilities, remember that it's important to keep your plans (short and long-term) transparent.

Not every product release can include each item from every wishlist. So Project Managers need to be realistic.

Short-term plans will likely be more detailed and outline plans in granular detail.

Long term plans often don't have this luxury.

Make these plans available to development teams. And be prepared to answer questions and concerns. And respond to input from the teams you are coordinating.

6. Assess product backlog and establish sprint plan

When you're establishing a Scrum approach across multiple teams you need to create one master product backlog for each team to extract their own backlog from.

The Product Owner should assume responsibility for managing and maintaining this backlog.

The project manager must oversee that this backlog is groomed regularly and thoroughly. This is to ensure that there are no requests sitting in silo.

Similarly, you must establish one clear plan which all parties adhere to. Clearly laying out the sprint plan, schedule for sprint retrospectives and testing.

Make sure none of the teams you are coordinating are rolling out their own diluted version of this. Success in Scrum depends on a single strategy for product delivery and transparency is crucial. The agile framework is built on unified thinking, transparent collaboration and constant communication. It requires discipline.

7. The technology enablers

What applications and platforms are your multiple teams currently using to track their workload and deadlines?

You may find that there are a mix of products and methods being used currently. But there is a need here for consistency.

Think about the technology you need to put in place to track progress and status of product and subsystem delivery.

If you already have something in place, is this easily rolled out to include disparate teams?

It may be that you need to implement a **synchronization tool** (like this one) to make sure everyone's workflows are aligned.

Another important technology consideration for multiple teams is **communication platforms**.

To succeed in agile working it is essential to move away from email and instant messaging conversations to a mechanism that enables you to view how the teams are collaborating and audit progress.

Note: *Issue trackers are perfect for this. However, if you're working with multiple systems, you will need a synchronization tool like Exalate. It'll help you keep all information up-to-date and in sync while the teams keep on working in their own environment.*

8. Offer training and support

An essential part of the collaboration process between multiple teams is offering ongoing support and enablement.

Make sure your teams understand what vehicles are available for them to access relevant information or ask questions.

9. Check back and make changes

Like any effective project management process, ensure that you inspect your agile collaboration processes regularly. And adapt as necessary.

If something isn't working, don't hesitate to change it. As long as any changes are communicated clearly and concisely to those who are affected.

Need for change can be a good indicator that progress is being made.

Suggest process inspections once per sprint.

Summary

The agile methodology was established to help businesses accommodate the changing needs in software development.

By introducing a unified set of processes and a focus on communication and collaboration, Project Managers are able to get better visibility of all aspects of the development project. And make adjustments to keep on improving.

This fluid approach allows better responses to market and customer influences. And it reduces time and resource being wasted on unnecessary project components and tasks.

Implementing an agile framework is not a simple task.

For Project Managers it requires much thought, preparation and support in order to be done effectively.

There also needs to be a consistent focus on improved communication and collaborative working. But the benefits for better software development and finer grained control are without question.