

Teams

This publication forms part of Emergn's Learning services. Details of this and Emergn courses, pathways, programs, and other learning products can be obtained on the Emergn website at <https://www.emergn.com> or contact us for a brochure – telephone UK +44 (0)808 189 2043 or US +1 617 482 0830; email info@emergn.com.

Emergn Ltd.
20 Harcourt Street
Dublin, D02 H364
Ireland

Emergn Inc.
190 High St, Floor 4
Boston, MA 02110
USA

First published 2011, revised 2023 - printed 15 November 2024 (version 3.0)
Copyright © 2011 – 2024

Emergn All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, transmitted or utilised in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without written permission from the publisher.

Emergn learning materials may also be made available in electronic formats for use by students of Emergn and its partners. All rights, including copyright and related rights and database rights, in electronic learning materials and their contents are owned by or licensed to Emergn, or otherwise used by Emergn as permitted by applicable law. In using electronic learning materials, you agree that your use will be solely for the purposes of following an Emergn course, pathway or program of study, or otherwise as licensed by Emergn or its assigns. Except as permitted above, you undertake not to copy, store in any medium (including electronic storage or use in a website), distribute, transmit or retransmit, broadcast, modify, or show in public such electronic materials in whole or in part without the prior written consent of Emergn or in accordance with the Copyright and Related Rights Act 2000 and European Communities (Copyright and Related Rights) Regulations 2004. Edited and designed by Emergn.

Printed and bound in the United Kingdom by Apple Capital Print.



CONTENTS

Introduction 1

1 The purpose of teams 2

2 What is a team? 7

2.1. Team features 9

2.2. Teams build over time 10

3 Teams don't work 12

3.1. Improperly bounded teams 12

3.2. Big problems require big teams 15

4 Solutions 22

4.1. How to deal with the big problem 22

4.2. A consequential purpose 24

4.3. Autonomy and self-organisation 27

4.4. Bringing the outside in 28

4.5. Scrum in action 31

5 Conclusion 33

Bibliography 34



INTRODUCTION

"Coming together is a beginning; keeping together is progress; working together is success."

Henry Ford

In this session we will discuss the working unit with which you are most likely to be familiar: teams. We will examine the rationale for using and not using a team; common dysfunctions of team-working and the methods and tools that will help you build a team that is set for success.

At the end of this session, the student will have an appreciation of:

1. When a team is the right tool for the job.
2. How to increase the probability that a team will be effective.
3. Why the majority of teams aren't working today.
4. What attributes a team needs to in order to be effective.
5. What stops a team from being effective.

1

THE PURPOSE OF TEAMS

There's hardly a job description in the modern world which doesn't demand 'team focus' or a 'team player'. In fact, so much of our working lives are geared around teams that it's worth asking whether we really need them.

Why bother?

When Michelangelo painted the Sistine Chapel he did it on his own. He didn't bother with a planning meeting and a progress chart and a review afterwards. There wasn't a constant stream of progress checks by management. Right?

Ummm. Wrong actually.



Figure 1. Sistine Chapel

Today's best estimates suggest that Michelangelo had help from 16 people in painting the chapel. Although he liked to foster an image of himself 'working alone without anyone to grind the colours', he needed help from experts – he tried two different types of scaffold-builders, he called in painters who had done frescoes before, and even a mould expert to explain what was wrong with the lime. It was a four-year project, and in that time he was constantly being interrupted by visits from the Pope to see how things were getting on. The Pope changed the deadline whenever he could, nagged about extras (how about some gold?), and got into performance related penalties when Michelangelo demurred (I'll have you thrown off the scaffold).

The point is not that Michelangelo was running a brilliant team, but that he needed one. Despite the myth, despite the ego, the problem of the Sistine Chapel was simply too big for one man, even for one genius, to solve on his own.

**Teams become important when:**

- **intense creativity is required**
An individual can come up with some brilliant ideas, but only a team can build on the idea quickly, solving challenges to bring the idea to reality.
- **fast learning is essential**
One person can study alone, but a team permits different members to become experts in specific areas.
- **many people need to buy in to the work/idea**
Someone – often a leader – can have a great vision, but in order to move everyone with her, she needs to ensure wider engagement. People working in teams around a shared objective become ambassadors for the idea.
- **different specialisms are required to solve the problem**
When the problem is larger or more complex than any one person's knowledge, different experts need to work together.

Activity 1: The right tool for the job?

We're pre-programmed to believe that teams are the way to solve any problems larger than the individual. But it doesn't always have to be a 'team'. You could have one person in charge, telling various individuals what to do and then putting their work together. This is not a team.

This activity should take around 10 minutes.

Which solution is right for your problem? And what factors influence your choice?

Gather the members of a team in which you work and discuss the following questions:

What do you feel determines when to select a team as the right tool for the job?

What sort of tasks or problems does a team work on most effectively?

How does this compare with our list above?

A team needs to serve something external

Perhaps because it seems so obvious, it can be forgotten that the purpose of a team must always be built around a problem. A problem requires the team to have an external focus. Michelangelo got so mesmerised by his own painting that he was in danger of forgetting the customer – the Pope. Modern customers won't have teams who aren't hitting the brief thrown to their deaths, but teams who don't satisfy the customer may find themselves out of a job.

Teams that are inwardly focused, fail.

Teams that have their eyes on the larger objective, succeed.



Figure 2. Pope Paul III

CASE STUDY: Parkside Hospital: A failure to focus on the patients

A major New Jersey Hospital (Parkside Hospital) began a large project on work life. Part of the project focused on improving the satisfaction and performance of nursing teams.

Turnover and conflicts in the teams were high – which everyone suspected was having a negative impact on performance, as well as on how the nursing teams felt.



Figure 3. Parkside Hospital nurses

Significant investment in time, energy and money went in to training team members in skills such as problem solving, communication, supervision, group decision-making and conflict resolution. They worked on understanding teammates' feelings and viewpoints and coming to consensus on what the team really wanted to achieve.

The good news: during the course of the project, the unit did improve communication and increase problem-

solving capabilities. People reported feeling more pride in their work, individuals learned interpersonal skills, and the number of work conflicts decreased.

The bad news: despite all the positive feedback, the nursing teams felt the changes would be short lived. Even worse, it became clear from the nursing audit data that there was little clear proof of any improvements in the performance of the nursing teams where it really mattered – making patients better.

The point? The exercise was so focused on the internal dynamics of the team that it failed to look externally to the overall goal – helping patients.



Is a team the right tool for the job?

We're going to look in our next section at how we define a team. Here, it's worth saying that a team is not always the panacea it is sometimes presented as.

A traditional unit of people led by a single leader with individuals taking responsibility for specific areas is often the perfect group to tackle a problem. We wouldn't call such a group a team according to the fairly strict definition we will discuss below.

For example: a group of senior department heads who meet once a month to exchange top-line figures is not a team. A dozen people who have similar background and skills are not a team, even if they are grouped together as a 'department' by the company.

If you or your organisation is sure a team is right for the job, then you better pay some attention to what this 'team' business looks like. Below is a quick check list of which working solution is best to tackle differing types of problem.

When to work alone or in other types of group	When to build teams
For simple tasks or 'puzzles'	For highly complex tasks or 'problems'
When co-operation is sufficient	When consensus decisions are essential
When minimum discretion is required	When there is a high level of choice and uncertainty
When a fast decision is needed	When high commitment is needed
When few competencies are required	When a broad range of competencies is required
When the organisation credits individuals for operational outputs	When the organisation rewards team results

Figure 4. When to build a team



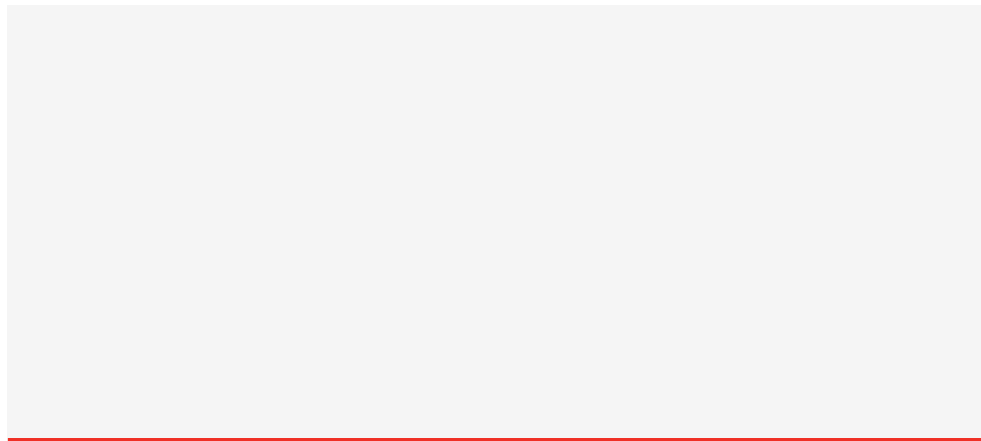
Activity 2: Defining a team

Think back to Activity 1. Did you notice that we instructed you to work with other people – in a team? Did you? We identified the type of task we were giving you, and realised that it was more appropriate to use multiple people, together.

This activity should take 10 minutes.

Gather the same group of people that you met with last time and define what you all feel are the key attributes of a team.

We'll see later that authors that have written on this subject over the last 20 or 30 years largely agree with each other. How much do you and the rest of the team agree?



When teams work they are able to produce something that we might as well call 'magic'. This goes far beyond the functionality of a group of collaborative experts, and enters into an unpredictable and emotive world of creative genius. The team of engineers who designed an emergency protocol to shut down the Command Module in Apollo 13; the team who designed the first Macintosh computer; or the team who made The Lord of the Rings movies in New Zealand. These were teams working as more than the sum of their constituent parts, teams who produced magical results. It is why, for all the difficulties and failures we will examine, it is worth striving to harness the potential of teams.



2

WHAT IS A TEAM?

"A team is a group of people working together to achieve a common purpose for which they hold themselves mutually accountable."

Peter Scholtes, *The Team Handbook*

"A small number of people with complementary skills who are committed to a common purpose, set of performance goals, and approach for which they hold themselves mutually accountable."

Jon Katzenbach, *The Discipline of Teams*

There are many subjects that cause divided opinions: are cats nicer than dogs? Should you make an omelette using butter or oil? But the theory of teams is not one of them.

Of the hundreds of authors who have written on the subject, few disagree. We have selected four representative 'business gurus', generally accepted as experts and who have recently written well-known books on the subject. We summarise the most salient points of their definition of 'teams' in the following table, before going on to examine each point in more detail. Note the similarities in their views, even when they use slightly different terms.

Hackman - Senior Leadership Teams: What It Takes to Make Them Great

- Have to be bounded
- Have a compelling direction
- Need enabling structures
- Need a supportive organisation (external)
- Need expert coaching

Katzenbach - The Discipline of Teams and The Wisdom of Teams

- Have a common commitment and purpose
- Have a common set of performance goals
- Have complementary skills
- Are small in numbers
- Be mutually accountable
- Have a common working approach

Larman - Scaling Lean & Agile Development

- Have a shared responsibility
- Have a set of working agreements
- Have a shared work product
- Have interdependent work
- Be responsible for managing the outside-the-team relationships
- Have distributed leadership

Sholtes et al - The Team Handbook, 3rd Edition

- Have a shared responsibility for outputs and results
- Have a commitment to a common approach to working together
- Have a shared work product
- Have tasks that are interdependent
- Collectively manage their relationships across organisational boundaries

Figure 5. Features of a team



2.1. Team features

Having a shared purpose

Sometimes this means a shared work product: the literal production of a working solution or prototype. It doesn't have to be tangible, the team could just as easily be working on a relationship or a roadmap, but each team member needs to be contributing towards the clearly-defined, shared purpose.

A compelling direction

Compelling refers to both the importance of the task – it must be real and matter to the organisation's overall success – and that it should be timely: teams work best with a measure of urgency. The direction covers not only 'where' the team is going, but 'why'. Note that this is not the same as a leader with a vision. Indeed, a key feature of teams is that leadership is distributed throughout the team.

Interdependent work / complementary skills

A direct effect from one team member's expertise or work should be visible within the shared work as a whole. In general, teams involve collaboration across multiple skill sets and functions to achieve the desired result.

A shared responsibility

Team members are held accountable as a group and individually for the shared work. It is this that allows teams to become more than the sum of their parts. It also means that teams must have a measure of control to be able to successfully achieve the outcomes.

Measured performance goals

A team must create a concise series of performance measures which will allow them to judge their progress and ultimate success or failure. Without being outcome-orientated, a team cannot achieve. It is important that when considering performance goals the team gives thought to the leading indicators that contribute to outcome, and not just the outcome measures. It is great to have a clear outcome in mind, but we really need to tune performance around metrics that help us understand progress and direction too.

Activity 3: What's missing from your team?

This activity should take 30 minutes.

Gather your team together and show them the list of team features above.

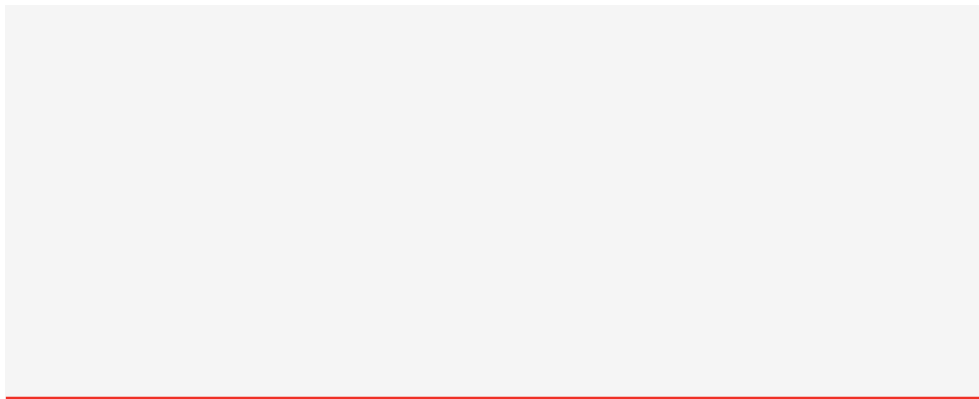
Are any of these features missing from your team, or could you improve any that do exist?

Select one of the features and identify the smallest possible change you could make that would make an improvement.

What impact would the change you're suggesting bring to the people on the team, the team itself, its customer and the organisation?

If you've reached consensus, why not go ahead and make that change?

Bear in mind that making a change will not only affect your team, but also people that you might consider external to your team. Do involve them in active discussion before making the change – they may have a very different viewpoint which you should take into consideration.



2.2. Teams build over time

According to the National Transportation Safety Board database, 73% of incidents occurred on a crew's first day of flying together – and 44% of those took place on a crew's first flight. A NASA study found that fatigued crews who had a history of working together made about half as many errors as crews composed of rested pilots who had not flown together before.



Figure 6. Miracle on the Hudson



Teams need a chance to become teams and build their performance over time. Once settled, it is best to allow them to continue together – something which even the best organisations often struggle to appreciate.

A famous model of how teams evolve over time was produced by Dr Bruce Tuckman in 1965.

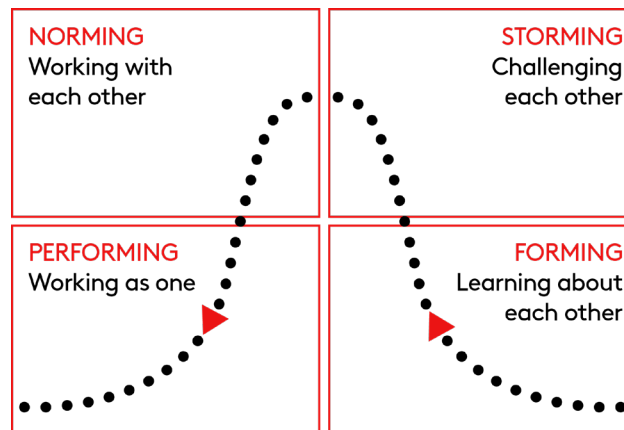


Figure 7. Tuckman Model

Forming: there is little agreement except through the leader's specific exercise of authority. Processes are often ignored as members try to discover their responsibilities, role and question the team's purpose and objectives.

Storming: decisions are made with a great deal of difficulty, a process which can encourage mini cliques, factions and power struggles within the group as members attempt to establish their positions in relation to one another.

Norming: roles and responsibilities are clear and accepted allowing big decisions to be taken jointly and smaller decisions to be delegated to individuals or sub-groups. Consensus leads to commitment and a general respect for the team and leadership.

Performing: the team has a shared vision, focus and purpose, and works with a high degree of autonomy. When disagreements occur they are resolved within the team, which makes progress on goals while also attending to relationship, style and process issues.

Teams need time to run through this cycle in order to achieve their full potential.

The principles discussed above are true for any team, from senior executives to product designers. Some types of task or work require a team method of working more than others – software development comes close to heading the list.

You undoubtedly already know how many problems occur if all design attributes have to be decided at the start of a project and presented as written requirements, or the delays that can occur if testing is held off until the very end of a project. New methods of working were designed to tackle such problems.

Given the challenges faced in software development, a dynamic, collaborative response is required – and that is a response best delivered by a team.



3 TEAMS DON'T WORK

"Genuine teamwork in most organisations remains as elusive as it has ever been."

Patrick Lencioni, Five Dysfunctions of a Team

We said above that pretty much everyone agreed on the major attributes of working in a team. Probably you nodded your head at each of the points. Yes. Uh huh. Of course. Well obviously!

And yet although 'team' looks simple on paper, it often doesn't work in practice.

How many teams have you been on that left you feeling energised, excited and knowing you were achieving valuable work? How many left you feeling frustrated? Did you dread team meetings? Did you see your hard work canned when a project was cancelled, or delayed, or perhaps even worse launched with a shed-load of problems that made you ashamed of it?

If you recognise the first question, and can't understand where the others are coming from then you're in a lucky minority.

When Tom De Marco and Timothy Lister began analysing IT development project histories, they discovered that of the 500 they studied, 15% failed, a number that increased to 25% for large projects. The reason? It was not technological failure that sank the projects. Instead, it was a combination of 'soft' problems: communication failure, high staff turnover, lack of motivation... In other words – the project didn't fail, the team did.

Why do teams fail?

There are a lot of factors that go into such a statement, from the way humans behave to the external support systems (or lack thereof) set up by the organisation. Here we are going to examine some of the most common and important.

3.1. Improperly bounded teams

Richard Hackman interviewed and analysed 120 senior teams. All of the teams said that they had set unambiguous boundaries, yet when asked one of the simplest questions imaginable 'Who is on your team?', fewer than 10% could agree on who the team members were.



Activity 4: Who's on your team?

Your turn! Who's on your team then? Write down the members.

Commentary:

Done? Great. Hold on a minute though; which one of the teams that you're in did you list the membership of? If you were to ask other members, do you think they would come up with an identical list?

Why do you think we're suggesting that the lists might not be identical? How often in your career have you only been on one team? What effects has multiple team membership caused, in your opinion?

It's interesting to note how loosely bounded the majority of teams are. As we progress through the remainder of this section, you'll discover that the majority of modern organisations are actually increasing the likelihood that their teams will fail because of their own structure.

Deciding who should be on a team is crucial – but often the team is formed before there is clarity about which skill sets are required. Teams need to be created with regard to the task in hand and that means being ruthless about selecting those with the right skills or the potential to acquire them. People should not be on teams 'because they're lovely', 'because they would be offended otherwise', or 'because they've always been involved in Project X'. Nor should they be on the team if they are unable to commit to it. That's a tough call, which might mean excluding someone with the perfect skills but who is already working on three other projects; or someone who knows a great deal about the subject but who hates teamwork and acts as a 'derailer' of other people's ideas in team meetings.



"Got to admit, as far as mission statements go, it's pretty damn bold."

Figure 8. Masters of the Universe

Even more commonly, teams don't agree on what they are supposed to be doing. It's easy to parrot a 'mission statement', it's much harder to ensure that this is a challenging purpose, focused on achieving actual, measurable results and that every team member understands how that will translate into their day to day activities.

Both of these elements are part of 'bounding' a team.

Let's take a Formula One racing car team as an example. Who is the team? Is it the driver? The pit crew? The design team for the car? The guys gaining sponsorship and funding?

If you made the team purpose 'winning the race' – you might have to include all of those people in your team, because the purpose has become so large that it incorporates the total value flow.

To create a functioning team, you need to define your objective better. Perhaps, 'those responsible for getting the car over the finish line on the day of the race'. That would exclude those gaining the funding and building the car – their roles are crucial, but by the day of the race itself their job is done.

Once the objective is set the team becomes properly bounded – only those necessary to achieve that specific objective can be included: the pit crew and the driver.

3.2. Big problems require big teams

It sounds sensible, doesn't it? If one rower can row a boat at 2 knots, then surely ten men can make it go at 20?

And yet 'common sense' is often wrong.

Here are the world record times over a 2000m distance.

Boat	World Record
Men's Single	6 mins 33 secs
Men's Pair	6 mins 42 secs
Men's Four	5 mins 36 secs
Men's Eight	5 mins 19 secs

Figure 9. Rowing World Record Times

As you can see, eight men do row faster than four – but not by much. That's a lot of effort that has gone into gaining 17 seconds. The physics is fairly simple – add an extra person, and although you gain their extra muscle power and force, you also gain their extra weight, which adds up to extra drag on the boat.

Still, you might think, at least we are getting faster – even if we're getting faster in a very expensive and wasteful fashion.

In the ancient world, trireme galleys were powered by oars rowed by up to 170 slaves. Tests have suggested that at their top 'ramming speed', (which we shall assume the slaves might be whipped into managing for the world rowing championships), they could reach 8 knots. Which means they would have travelled the 2000m in 8 mins 10 secs – a lot slower than our one-man boat.

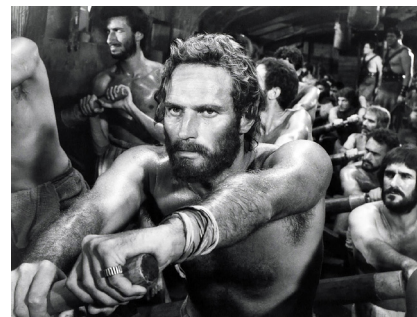


Figure 10. Ben Hur in a trireme galley

Adding resources is often seen as the most sensible, indeed, as the only way, to tackle a problem. The reality is that large groups come at an increased cost which can often make the project go slower, rather than faster.



Activity 5: Moving tables

This exercise requires a minimum of six people.

Object:

The aim of the exercise is to move a table from one side of a room to another in the fastest time possible. Ideally this will require moving a table (or similarly awkward object that is difficult for one person to move on their own) five metres or more.

Timekeeper/Facilitator:

Nominate someone in your group who will serve as the Timekeeper for the exercise and referee play. The Timekeeper should read the rules below to the group.

Rules:

1. All members of the team must touch the table at all times whilst moving the table.
2. Members of the team can only communicate how they will complete the task once the exercise has begun.
3. The Timekeeper cannot be part of the team moving the table.
4. The Timekeeper decides whether the task has been completed.

Task:

Complete the following task starting with a team size of one and repeating the task with an additional team member until all members of the group are included.

First, get the team to estimate how long they will take to complete the task. The Timekeeper should record the estimate in the table below.

The Timekeeper starts the exercise by counting down "3-2-1-go" and starts the timer.

The team moves the table from one side of the room to the other.

Once the table has been successfully moved the Timekeeper stops the timer and records the time taken to complete the task in the table below.

Team size	Estimated time	Actual time taken

**Discussion:**

At the end of the exercise discuss with the group the effect that adding more people to the team had on the overall performance. Did you see the improvements you expected? Did the performance improvements increase linearly as people were added?

Review:

Now that you have run the activity, reflect on the events leading up to it. What did you have to do to get a group of people together? How did you ensure that the group of people you had gathered understood what was required of them? Did you encounter any issues that you had to overcome?

How do these experiences reflect on what you expect it might take to create a team?

A large, empty rectangular box with a light gray background, intended for participants to take notes or reflect on the discussion. A solid red horizontal line is positioned at the bottom of the box.

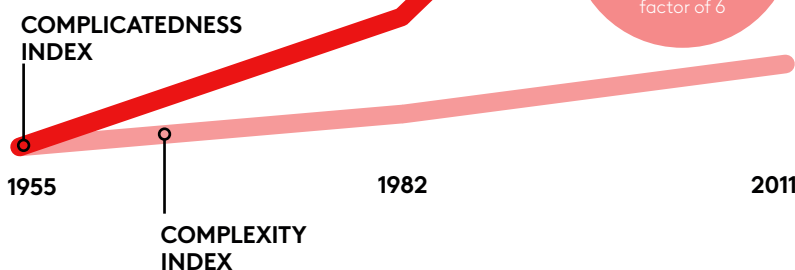
“Research consistently shows that teams underperform, despite all the extra resources they have. That’s because problems with coordination (caused by dependency) and motivation typically chip away at the benefits of collaboration.”

J Richard Hackman, Why Teams Don’t Work

As Jon Katzenbach writes in his book, *The Discipline of Teams*, ‘Virtually all effective teams we have met, read or heard about, or been members of have ranged between two and 25 people’. This is because the coordination costs between large groups of people quickly escalate to levels which are unsustainable. Because the group inevitably breaks down into smaller sub-groups, the risks increase that their goals will become misaligned, leading to wasted effort, or worse, impeding one another. This is often surrounded by an increasing number of procedures, layers of management, coordinating bodies making it increasingly hard to make decisions, as seen in the figure below.

The Cost of Inept Responses to Complexity

Since 1955, business complexity, as measured by the number of requirements companies have to satisfy, has risen steadily. To address each new requirement, companies typically set up a dedicated function and then create systems to coordinate it with other functions. That explains why organisational complicatedness (the number of procedures, vertical layers, interface structures, coordination bodies and decision approvals) has seen an even sharper increase. This complicatedness hurts productivity and employee engagement.



Source: *Smart Rules: Six Ways to Get People to Solve Problems Without You*

Figure 11. Complicatedness vs Complexity

CASE STUDY: Google takes it one step further

Google, under its co-founders Sergey Brin and Larry Page, structured the whole of the business around small, tightly focused teams. They decided that many small teams working at once on limited projects would be more effective than one large team tackling a mammoth project – and they turned out to be right.

Given the vast nature of its user base, Google has an unprecedented capacity to try out an idea, from a user interface to a wacky app, on a tiny proportion of its users and get massive feedback. That means teams who are throwing out new software every few weeks can refine, change, retest, progress or kill a project, and then move on to the next.

One success in ten tries is okay; one in five is superb. If a venture fails, the team moves to another urgent project. “If something is successful, you work it in, somehow,” Schmidt says. “If it fails, you leave [it].”

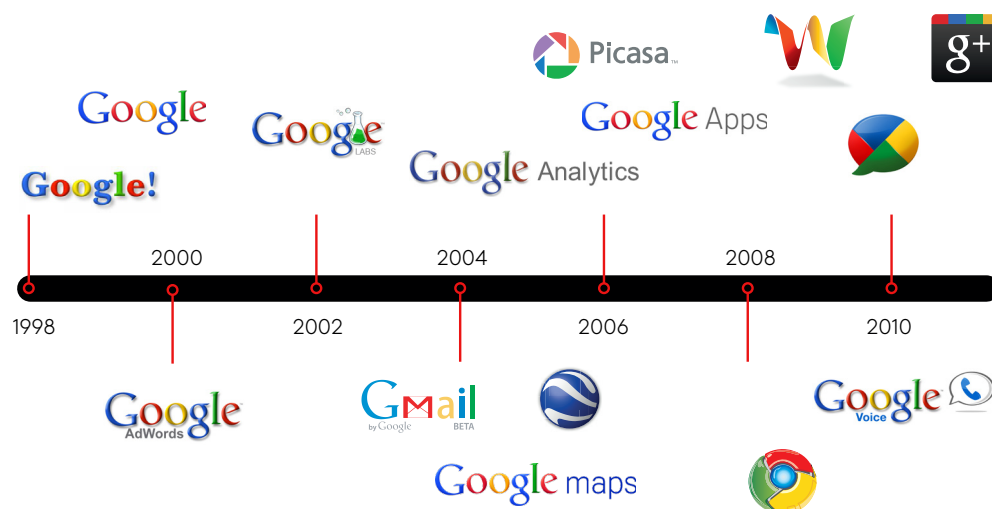


Figure 12. The Innovation of Google over the years

To capture ideas and find traction, an “ideas mailing list” was set up. Every time someone posts an idea, an email is sent to everyone on the list. Each email is half a pitch for an idea, half an invitation to join in developing it. These idea emails are responsible for much of the software on Google’s projects.

This approach, where the teams own idea generation and also own customer feedback to decide on success or failure, means that Google has fewer managers than most companies: one for every 20 employee, compared with one for as few as seven industry wide.

Keep the team small

The right response is often counter-intuitive to the common sense we feel we know. Fred Brooks wrote a sentence which was so appropriate it has been turned into a principle of software development, called Brooks's Law: 'adding manpower to a late software project makes it later.' (The Mythical Man-Month, 1975). As the graph below shows, more people on a project increases the time it will take because of the additional complex inter-relationships that must be managed.

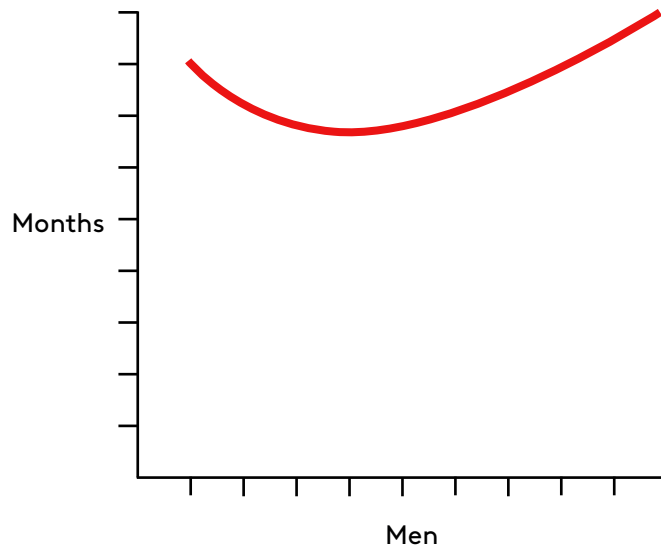


Figure 13. Shows time taken versus number of workers for tasks with complex inter-relationships like software

Many organisations are filled with people who have read this in dozens of text books and nodded sagely. Then those same people continue to create over-sized teams that fail and throw the wrong type of resources at a large project in the hope that this time it will somehow be different.

Consider our trireme galley filled with rowers. The designers of those galleys were highly skilled – they balanced a need for speed with a need for fighting (they wanted to mount bronze battering rams and later on guns). But it took a different kind of design altogether to allow ships to travel further at a lower manpower cost – a new type of rigging that allowed sailing ships to tack into the wind. Just so, freeing a team from the weight and baggage of being over-sized can permit a more creative, brilliant solution.



Activity 6: Size is everything

This activity should take 10 minutes and can be completed alone.

Go back to the list of people you decided were on your team in Activity 4 and consider the following questions.

Do you think your team is the right size for the job?

Would you be doing a better job if you had more people?

What about with fewer people on the team?

Which roles or skill sets are essential for the team to meet its purpose?

Which roles or skill sets are nice to have?

Could we break down the work, making it easier to allocate to differing teams?

4 SOLUTIONS

If teams are important, but they often don't work – what then?

Many of the problems we described are about how the team has been set up. We will examine these first before looking at ways of improving the functioning of the team itself.

4.1. How to deal with the big problem

Just because small teams are more effective than big ones doesn't mean that three people are all you need to run the organisation.

Often the problem is simply too big to be dealt with by only five to seven people, or the timescales are such that even if they could deal with it, the amount of work would take them several years.

So what do you do?

Let's start with what most companies do – which is to share the big problem amongst lots of poorly-defined teams. This is exactly how to ensure wasted effort. Even if you think you have split the problem, you'll find the problem has often been split sequentially – that is there will be groups of people waiting for a team to finish, and because they don't want to sit around idle, they'll start creating work which often clashes with the original team.

The trick – and it's not easy – is to split the problem into well-defined smaller pieces of work, each of which can be tackled by an independent team which is not reliant on sequential elements. For example – a small part of the problem would involve a team doing its own plan, build and testing, rather than a planning team, a build team and a test team.

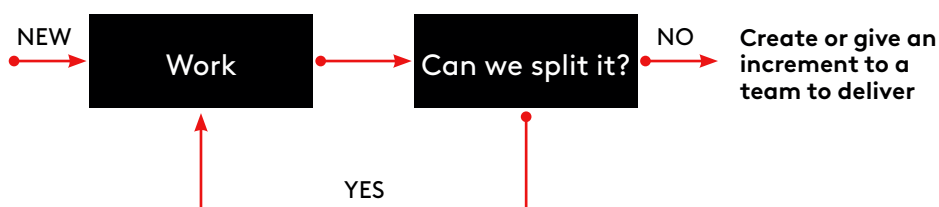


Figure 14. Splitting the problem



CASE STUDY: Heroku targeted teams

Heroku is a cloud-based platform that allows companies to scale and deploy their apps. It was set up as a small start up in 2007. One of the founders, Adam Wiggins, discussed managing team sizes at different stages: when a company is '2-4 guys/gals in someone's living room', there is no need for management; everyone is a generalist doing a little bit of everything and knowing what everyone else is doing.

When the company grows a bit – maybe to 9 employees, a single team is still OK, but a few bits of process are required to ensure people don't duplicate effort. Wiggins writes, 'Everyone is still a generalist, but the whole development team should be aligned behind a single goal (aka milestone) at a time.'

Eventually the company grows further. With 10-15 developers, daily standups start to become a pain. In programming, when a class grows too big, the answer is to break it down. Wiggins points out the same holds true for organisations – he calls the result 'targeted teams'.

In Heroku, they paid a great deal of attention to ensuring each team owned a clear vision and direction and could work with maximum autonomy. They tried to map the software architecture onto the team architecture. Their initial team breakdown looked like this:

- API - Owns our user-facing web app and the matching Heroku client gem.
- Data - Builds and runs our PostgreSQL-as-a-service database product.
- Ops - Shepherds and protects availability of the production system.
- Routing - Manages everything necessary to get HTTP requests routed to user web processes.
- Runtime - Handles packaging code for deploy and starting/stopping/managing user processes.

Wiggins sat down with each developer to work out which part of the system he/she was most passionate about working on. They tried out a role and then arranged a transfer for anyone dissatisfied there. He concludes, 'Let developers follow their own passions and they will gravitate toward the team where they will do the best work.'

With specialists beginning to appear as well as generalists, Wiggins fixed the team size at 3 – two developers (to ensure they could check code together) and one business-owner. Often these were generalists transferring to management and occasionally worked across two teams, while developers were focused solely on a single team. Wiggins finishes by commenting on the power of targeted teams, 'A few months after breaking into teams at Heroku, we had a day where three different teams were all releasing major new features. It's an incredible feeling.'



4.2. A consequential purpose

A team needs to know that their work will make a difference, that the sum of their activity (or lack of activity, if it comes to that) has consequences. As soon as teams have a real purpose, whose importance they understand, their work is driven in a compelling direction.

Sometimes this can throw up some very controversial issues. Functional 'teams', like a Project Management Office or a business analysis team do not have a consequential purpose. That is not to say their work isn't valuable to the company – it undoubtedly is – but the functional department is not directly responsible for revenue or innovation, it only contributes towards it. Splitting up functional teams and embedding individuals in real, cross-functional teams helps companies harness their employees' efforts more effectively.

Activity 7: Your team's purpose

Distribute a copy of the questions below to all of your team members. In Activity 8, you will be asked to arrange a meeting to discuss the answers as a group so ensure your team members keep their answers. Take 5 - 10 minutes to note down your own answers to the questions.

What's the purpose of your team?

What problem does it exist to solve?

Exactly how does that contribute to the organisation meeting its vision?

How will that be of significance to the organisation?

What would happen to the organisation if your team did NOT achieve its purpose?

How will you know when your team is done?

When does your team need to finish by? Why?

Will customers (external) notice something different?

CASE STUDY: Faster growth at Avery Dennison



Avery Dennison, an office products company, wished to achieve faster growth. It had a large project focused on this strategy with initiatives such as 'new technology investment' and 'market overview'.

There was no compelling direction and growth was correspondingly sluggish. Then the company began trialling a few small teams on specific sub-problems: 'secure one new order for a specialised product, refined in collaboration with one large customer, within 100 days'; 'sign up three retail chains and from that develop a methodology for moving into new distribution channels'; 'collaborate with three other suppliers to provide all the parts needed by a major customer'. The teams produced results fast. The idea of small focused teams was rolled out to the wider business and by the end of the first quarter the teams had produced \$8 million in new sales.

Note that the big problem does not vanish. The results from the focused team are fed back in to the larger problem. Gaining one customer is rarely the silver bullet to a company's woes or dreams, but it does provide a roadmap for how more customers might be acquired.

A little warning: the process of learning how to break down a problem is not always straightforward. It requires discipline and creativity in itself.

Once again, the thinking here is not dazzlingly original. Traditional Project Management methodologies, including Prince2 and PMBOK, recommend 'the breakdown of subdividing project deliverables and project work into smaller, more manageable components'. However, these methodologies focus on displaying progress within the project, not creating independent teams with limited but compelling goals.

A goal needs to be both challenging and realistic, while the team's actions must be consequential – that is, have a measurable impact on the goal, not just measure progress of an activity. In order to do so, more attention needs to be paid by senior management to the set-up of the team. Investing the time upfront in working through the purpose and direction sets a team up for success.



Activity 8: Getting the purpose right

Let's go back to your answers from Activity 7. Convene a team meeting – to do properly this activity will take at least 1 hour.

At the start of your team meeting, go through everyone's answers – perhaps jotting them down on a whiteboard. Make sure everyone's answers are heard.

Are the answers in roughly the same ballpark, or does it look as if you all work in different companies?

If it does, don't worry. This is quite normal! Now is the time to invite the group to try and agree upon a shared purpose. Simplify the questions:

What is the problem we are trying to solve?

How will it affect the company if we fail / if we succeed?

Now try and express the answers as a single purpose.

Once you have a purpose that everyone is happy with, check the following questions.

Is it compelling? Can it be measured? What's the time frame? Who's on the team?

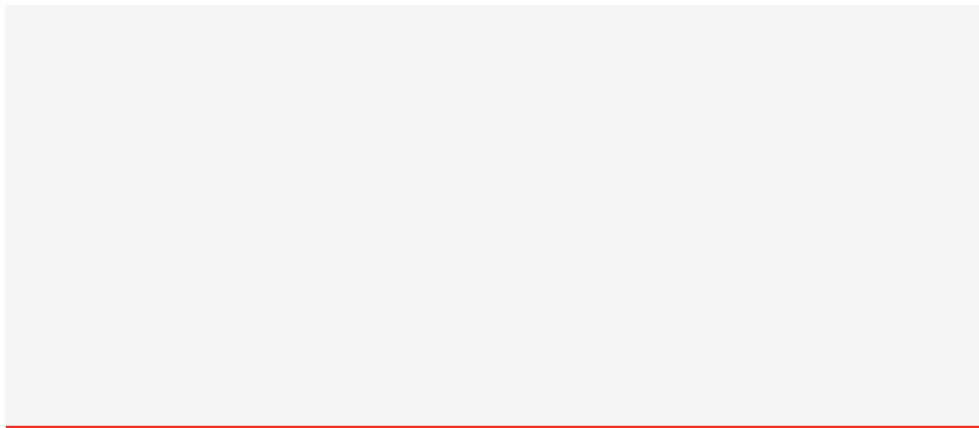
You may find that this activity leads you to consider how the team is structured.

To make your purpose compelling, for example, you may need to create a series of smaller sub-teams to deal with other parts of the original problem.

Could the problem be broken down further?

Could your team be smaller?

Could each sub-team still have a compelling purpose?





4.3. Autonomy and self-organisation

“Over the last several decades, the dysfunctions caused by managers over specifying and controlling aspects of a team’s work in real time have been amply demonstrated to hurt both people and organisations.”

J. Richard Hackman

We have mentioned the work that needs to go into setting up a team properly: far more effort, time and thought is required to properly define goals and bound teams than most managers give. The good news is that far less time, work and thought needs to be spent overseeing the team than is currently devoted to it.

Unfortunately, this is a switch that a lot of managers find hard to make.

Example:

A software development team knew that it needed to gain client feedback on each new piece of code it was turning out. The team’s manager insisted that nothing could be shown to the client until he had passed it personally. Since he was swiftly overloaded with work, very little was flowing out to the customer, which meant the team continued working in the dark, knowing full well that much of what they did would need extensive revision.

The productivity paradox

Self-directed employees achieve more than those who are passive – the more managers attempt to control their workforce in order to ensure productivity, the more passive employees become with the paradoxical result that productivity plummets.

Researchers at Cornell University studied 320 small businesses, half of which offered their workers autonomy, half of which relied on top-down direction. The businesses which offered autonomy grew at four times the rate and had one third of the staff turnover.

Self-organisation and autonomy does not mean that managers should simply pack up their desks and go home. It relates to the initial set-up work in properly bounding a team. Teams need to know what comes within their authority and what does not.

Is a team responsible for setting documentation standards for testing as well as coding? Having discovered a potential customer is interested in a specific feature, should the team go all out to develop it?

Without delineating the extent of a team’s sphere of authority and operation, the team will engage in what Don Reinertsen, in his book *Managing The Design Factory*, called running into ‘invisible electric fences’ – a painful, demotivating and inefficient experience.



The Nobel Prize-winner Ilya Prigogine stated that the boundary is what defines self, and only within this can any complex system become self-organising. Software development teams are an example of a complex system.

So – establish clear boundaries for the team, and then allow them to self-organise.

Once done, it is essential that managers trust their teams and resist the temptation to tamper.

4.4. Bringing the outside in

John Donne famously wrote: ‘no man is an island, entire of itself; every man is a piece of the continent, a part of the main’. The same is true for teams. We talk about autonomous teams, but of course, they do not exist in isolation. They are part of a larger company and there are dependencies and touch-points within the company that need to be thought about.

Imagine a software team, busy developing and churning out working features.

We have drawn attention to the dangers inherent in not involving the customer – what if you are developing features they don’t need and failing to develop their essential requirements? A constant feedback loop is important – but customers have their own businesses to run. If you are developing software to help heart-surgeons operate – it’s probably best if you don’t drag one of the best heart surgeons in the world over to your office for a weekly meeting. Scrum attempts to solve this problem by creating a ‘Product Owner’; literally someone who represents the customer and works within the team.

This is what we mean by ‘bringing the outside in’. An external dependency needs to be incorporated within the team.

But customers are not the only external dependency. Many teams send out a flow of information to a Project Management Office – who in turn report progress to senior management. Feedback flows back by the same method. Here is a potential block to efficient work flow – the more steps there are, the easier it is for delays and misinterpretations to occur. Can the project management function be brought within the team? Is there a role within the team for measuring and reporting that can flow directly out to senior management?

What other dependencies exist? Think not only of those to whom you report, but also those who do something for or bring information to you. The aim is to make the team as independent as is practically possible.

External dependencies can range from data gathering to testing, sales to maintenance, or documentation to integration.

The most important of these – the ones which are crucial for the team’s success, or where the relationship is unsatisfactory – can often be improved in some measure by being brought within the team. If a software development team finds itself constantly moaning about the idiocy of marketing who continually misrepresent their products, then it’s time to bring marketing within the development team.



CASE STUDY: Semco: truly autonomous teams

Richardo Semler took over his traditional family engineering company in the 1980s. One of his first actions was to fire 60% of all managers. Today, Semco has one of the flattest management structures in the world, because all employees share responsibility for many of the traditional 'management' tasks. These include cash flow statements and balance sheets (all employees are trained to interpret them), as well as organising their own hours and making many operational decisions. During the downturn in Brazil in the 1990s, the management methods at Semco managed to achieve a 65% reduction in inventories, a marked reduction in product delivery times and a product defects rate that fell to less than 1%. Such successes meant the company did not only survive, it grew.

Semco's teams are truly autonomous, employees have experience in numerous different operational sectors. They are able to make decisions without running them through layers of management, financial or resource decision-makers – because they own these responsibilities internally within the team.

Getting the mix right

As you've been reading this, no doubt you've spotted a potential flaw.

We know that teams perform most effectively when they are small. However, in order to be autonomous, a team needs to have many different skills – which naturally tends to grow the team. How do we bring the outside in without creating unwieldy teams?

The answer is that the individuals within the team have to release themselves from rigid roles. In a cross-functional team, it's rarely acceptable to say 'I don't do finance, I just code', or 'I don't understand integration, I just do design'. Instead, although expertise in certain areas remains important, the team as a whole takes responsibility for project management, reporting and budget control. Usually individuals embrace this whole-heartedly, sometimes training may be required to help take on new responsibilities.



Activity 9: Who do you need?

Try this exercise on your own. It should take 20 minutes.

Which external dependencies impact on your team?

Consider: competitors; market trends; customers/donors/members; regulatory bodies; senior management; other departments/teams.

Now ask where there is a block to the team's progress.

Is there a delay in getting information or decisions?

Do you feel the team's purpose is not fully understood?

Have you developed work only to discover it is in conflict with the aims of the dependency?

Look at the group where there is most frequently a block or a conflict that impacts on your team's work.

To what extent are the concerns and interests of this group represented or considered in your team?

List five practical activities which might help you better understand their concerns and motivations.

A large, empty rectangular box with a light gray background, intended for participants to write their answers or notes during the activity. It is positioned below the list of questions and above the footer.



4.5. Scrum in action

Scrum was developed specifically for the working methods of development teams. Though not the only Agile framework to improve teams, it is the most popular approach today for small teams and can be extremely effective with the solutions proposed in Sections 4.1 to 4.4.

To describe how Scrum works requires a longer treatment than can be given here – refer to the Scrum session to see its principles and processes.

4.1. Dealing with the big problem and 4.2. A consequential purpose

Scrum is designed to break large problems down. The backlog is made up of numerous ‘user stories’, each of which represents the smallest manageable component that can build up to a function.

For example:

‘As someone with a current account, I want to check my balance online.’

The backlog is prioritised, the team begins with the elements the customer is certain about, leaving vaguer ideas, or less essential parts until later. Each of these small features has a clear, consequential goal, which together are aligned to the overall goal. By the end of the Sprint, I will either be able to check my balance online, or I won’t.

The team signs on to deliver a given number of these features at the beginning of each Sprint and only working, demonstrable code is accepted at the end of the Sprint. Anything half done or not yet working is returned to the backlog. Thus, not only the overall project has a consequential goal, ‘set up online banking for the bank’s customers’; but every Sprint also has a consequential goal, ‘any user will be able to check their balance’.

4.3. Autonomy and self-organising

The team is self-organising because it chooses what it can or cannot commit to at the start of each Sprint, it measures progress in daily Scrums, and it reviews the working method in a Retrospective at the end of each Sprint. This process helps the team maintain discipline and is designed to stop the legendary ‘death march’ at the end of a traditional IT project, since the team controls its own progress and has continual feedback between senior management and customer.

4.4 Bringing the outside in

Scrum focuses on the customer by creating the role of ‘Product Owner’ designed to bring the customer’s concerns to the heart of the team. In Scrum teams it is usual to also own testing, integration, and project management within the team.

A word of warning here – although the role of ‘Product Owner’ exists, it remains essential that Scrum teams do not forget they have a collective responsibility to bear the customer in mind. Remember the dictum stated at the beginning of this section: teams that are externally focused succeed; teams that are inwardly focused fail.



The Emergn View

If ever an annoying phrase abounded in management text books it's this: 'It's not rocket science'.

Effective team management is not rocket science. For a start, rocket scientists have a hard time getting it right.

Setting the correct shared objective is not an easy matter. It means making choices, and choices mean letting go of all the things we're not going to do. It involves everyone committing when they may have very different internal goals, motivations and opinions.

It is also high risk.

Identify an objective wrongly and your team will be pulling in the wrong direction. Problems will get worse. It could be some time before you are able to identify what's wrong and correct matters accordingly.

This is the strength of having teams focused in short bursts towards very specific goals. It is also why it is harder to do this with larger 'strategic' problems. The answer is to pay attention to the process of the team's working – every team, from a development team to a senior executive team, needs to set its objectives and measures and review them at regular intervals. Remember – objectives should be externally-focused: the goal must contribute to the overall value flow for the company and customers.



5 CONCLUSION

In this session we have looked at why teams can be helpful, where and how they succeed, what are the key barriers to their success, and some of the best solutions to common problems.

Learning outcomes

Now that you have completed this session, you will have an appreciation of:

When a team is the right tool for the job

- Intense creativity or fast learning for highly complex problems
- Many people need to buy in - consensus is important
- Different specialisms and broad competencies are required
- When there is a high level of choice and uncertainty
- When high commitment is needed

What attributes a team needs to be effective

- Shared purpose in a compelling direction focused on the external
- Interdependent roles and shared responsibility supported by organisational structure
- Measured performance goals that assess team not individual

What is stopping a team from being effective

- Improperly bounded in size or purpose
- Insufficient time together
- Incorrect skill set

Why the majority of teams aren't working today

- Too large – the number of interactions becomes unmanageable
- Unclear purpose or misaligned to the external objective
- Not provided with autonomy or resource to complete the mission

How to increase the probability that a team will be effective

- Split big problems; assign small teams
- Incorporate external dependencies within the team
- Support and promote autonomy
- Provide a consequential purpose

BIBLIOGRAPHY

- Ancona, D., Bresman, H., 2007.** X-Teams: How to Build Teams That Lead, Innovate and Succeed. Harvard Business School Press.
- Appelo, J., 2010.** Management 3.0: Leading Agile Developers, Developing Agile Leaders. Addison Wesley.
- Brooks, F., 1995.** The Mythical Man Month and Other Essays on Software Engineering. Anniversary Edition. Addison Wesley.
- DeMarco, T., Lister, T., 2009.** Peopleware: Productive Projects and Teams. 2nd Edition. Dorset House Publishing.
- Denning, S., 2010.** The Leader's Guide To Radical Management: Reinventing the Workplace for the 21st Century. Jossey Bass.
- Donne, J., 2001.** Complete Poetry and Selected Prose of John Donne. Ed Edition. Modern Library Inc; Modern Library Pbk.
- Forbes, 2005.** Google Thinks Small. [online] Available at: <<http://www.forbes.com/global/2005/1114/054A.html>>. [Accessed 30 July 2012].
- Hackman J., Nunes D., Burruss J., Wageman R., 2008.** Senior Leadership Teams: What It Takes to Make Them Great (Leadership for the Common Good). Harvard Business School Press.
- Hardy, Q., 2005.** Google Thinks Small. [online] Available at:<<http://www.forbes.com/global/2005/1114/054A.html>> [Accessed 01 November 2011].
- Harvard Business Review, 2011.** Smart Rules: Six Ways to Get People to Solve Problems Without You. [online] Available at: <<http://hbr.org/2011/09/smart-rules-six-ways-to-get-people-to-solve-problems-without-you/ar/1>>. [Accessed 02 September 2011].
- Harvard Business Review, 2003.** Why Good Projects Fail Anyway. [online] Available at: <<http://hbr.org/2003/09/why-good-projects-fail-anyway/ar/1>> [Accessed 02 September 2011].
- Harvard Business Review, 2009.** Why Teams Don't Work, An Interview With J. Richard Hackman. [online] Available at:<<http://hbr.org/2009/05/why-teams-dont-work/ar/1>> [Accessed 22 August 2011].
- Katzenbach, J., 2001.** The Discipline of Teams: A Mindbook-workbook for Delivering Small Group Performance. John Wiley & Sons.
- Katzenbach, J., Smith, D., 2005.** The Wisdom of Teams - Creating the High Performance Organisation. European Version. McGraw-Hill Professional.
- Larman, C., Vodde, B., 2008.** Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum. Addison Wesley.



- Lencioni, P.**, 2002. The Five Dysfunctions of a Team: A Leadership Fable. Jossey Bass.
- Pink, D.**, 2011. **Drive**: The Surprising Truth About What Motivates Us. Canongate Books Ltd.
- Reinertsen, D.**, 1998. Managing The Design Factory: A Product Developers Tool Kit. Simon & Schuster Ltd.
- Scholtes, P., Joiner, B., Streidel, B.**, 2003. The Team Handbook. 3rd Edition. Joiner/Oriel Inc
- Semler, R.**, 2001. Maverick!: The Success Story Behind the World's Most Unusual Workplace. Reissue Edition. Random House Business.
- U.S. Marine Corps**, 1996. Marine Corp Doctrine Publication 6, Command and Control. United States Government.
- Wiggins, A.**, 2011. How To Scale a Development Team. [blog] 28 April, Available at: <http://adam.heroku.com/past/2011/4/28/scaling_a_development_team/>. [Accessed 17 October 2012].



For more information
about **Teams** or our other
Emergn session books,
please contact us or visit
emergn.com

US +1 617 482 0830

UK +44 808 189 2043

Email info@emergn.com
