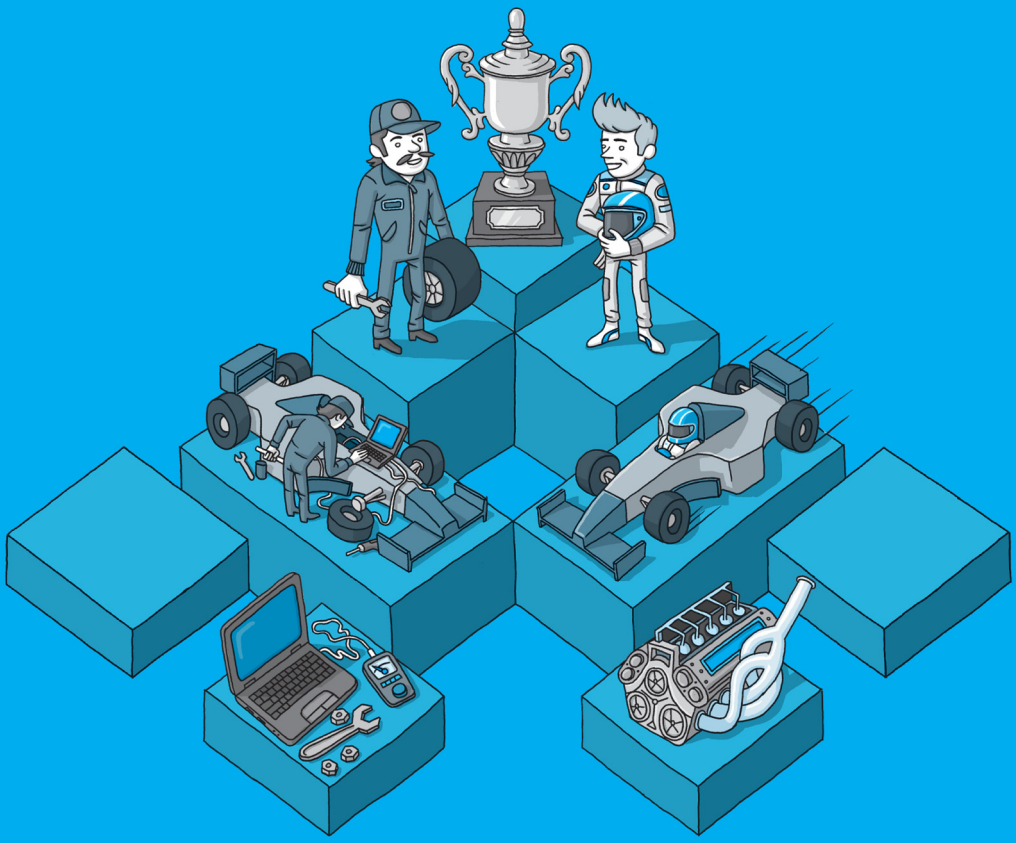
GOJKO ADZIC

IMPACT MAPPING

MAKING A BIG IMPACT WITH SOFTWARE PRODUCTS AND PROJECTS



foreword by **Tom Poppendieck**

Impact Mapping

Making a big impact with software products and projects

Gojko Adzic

This book is for sale at
http://leanpub.com/impact-mapping

This version was published on 2014-12-04

Tweet This Book!

Please help Gojko Adzic by spreading the word about this book on Twitter!

The suggested hashtag for this book is #impactmapping.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

https://twitter.com/search?q=#impactmapping

Also By Gojko Adzic

Fifty Quick Ideas to Improve your User Stories

Fifty Quick Ideas to Improve Your Tests

Contents

Foreword

With few exceptions, software development work has long been separate from other parts of the organizations it supports. The rest of the organization did not understand software and too often, the development organization was not tightly attuned to the business of the rest of the company. Difficulty communicating on the one hand led far too often to building the wrong thing, or at best, not quite the right thing and on the other hand, led to wasteful management and governance practices. Agile approaches have helped here by enabling rapid feedback cycles to correct mistakes before there is no budget left to do so.

We need an approach that engages development organizations as full partners with other disciplines in creating delighted customers which leads to business success. Joint engagement needs to be based on efficient and effective communication, but the disparate perspectives and even vocabularies dominating each discipline make unambiguous communication difficult. What is needed is a way to visualize the problem we are collaborating to solve in a way that allows people contributing from their individual discipline's perspectives to be talking about the same thing. Agile approaches use 'working software' as this visualization by rapidly implementing small collections of user stories which non-software disciplines can provide feedback

on. However, while working software can validate choices, it does not help a lot with discovering and selecting those sets of user stories whose implementation will delight the people who will use the solution to achieve the overall purpose.

Many of the barriers to effective collaboration arise from unshared, unexamined, unproved assumptions. Each discipline has different sets of assumptions. If these can be made explicit and examined and validated, it will be possible to get to better solutions faster. This is exactly where impact maps are helpful as they lay out visually the WHY, WHO, HOW and WHAT of the problem we are confronting.

Like highway maps that show towns and cities and the roads connecting them, impact maps lay out what we will build and how these are connected to ways we will assist the people who will use the solution. The primary purpose of a highway roadmap is not to provide detailed information about the towns and cities but rather to make explicit the connections among them; their secondary purpose is to help us identify alternate routes.

Discussions about the nodes of the impact maps and the assumptions which connect them can effectively engage all the disciplines in discovering innovative, efficient solutions. They provide an explicit context for making decisions and exposing uncertainty, which can then be addressed by doing experiments. Like closed roads or roads under construction, sometimes assumptions are not viable or are simply wrong. It is called a highway map, not a destination

map. What your map is really assisting with is visualizing first provable and later validated assumptions (roads).

We are seeing fundamental shifts in mind-sets from push to pull and correspondingly a shift from directive control to adaptive control. Push means people need to be told what to do; pull starts with a problem or opportunity and challenges a cross-disciplinary group to effectively understand and address the issues. This is a fundamental shift from a focus on 'manufacturing' what a customer orders to a focus on collaborating with other parties to achieve a purpose. This requires a shift from extrinsic motivation (push) to intrinsic (self) motivation (pull), and as Dan Pink so elegantly explains, intrinsic motivation is driven by autonomy, mastery, and purpose.

An autonomous group which collectively includes competence in the requisite disciplines is the only effective way to efficiently achieve the kinds of goals we attempt to achieve today. However, groups are not effective teams unless they have a shared target or goal. Shared goals arise from shared stories. Storytelling, or more deeply, shared story creation or discovery is what is going on here. An impact map is also a storyboard of our conceptual understanding of how to address a goal or target. This is too important to be left to a 'customer' or 'product owner' to push upon the team.

The impact map approach helps you to focus on understanding and evolving a response to even 'wicked problems' which increasingly are what we confront these days. This book advocates relentless re-visitation of every ele-

ment and validation of assumptions, which seems exactly what we need today. Even the initially selected impact or effect or goal or purpose is based on assumptions about the situation of the organization and the market, so it needs to be validated along with all the steps to address it.

Design is about finding and trying possible solutions that might cause the desired impact. The critical component is neither the cause nor the effect but rather proving the validity of the assumption that links them. Assumptions validated by consistent data from actual experiments enable the creation of real value. An impact map connects candidate causes with desired effects. It is a map of assumptions connecting causes and effects. It helps you to find the right questions, which is much more difficult than finding good answers.

– Tom Poppendieck

Introduction

After nine years of work and billions of pounds already spent, the UK government recently abandoned a doomed IT project because 'it wasn't essential'. Similar examples of less epic proportions are all around us. Commercial organisations across the European Union lost 142 billion EUR on failed IT projects in 2004 alone, mostly because of poor alignment with business objectives or business strategies becoming obsolete during delivery. This is roughly the cost of the International Space Station programme, including all flights, or almost twice the cost of the entire Apollo programme, which achieved six manned landings on the Moon.

Software is everywhere today, but countless software products and projects die a slow death without ever making any impact. The result is a tremendous amount of time and money wasted due to wrong assumptions, lack of focus, poor communication of objectives, lack of understanding and misalignment with overall goals. There has to be a better way to deliver!

This handbook is a practical guide to impact mapping, a simple yet incredibly effective method for collaborative strategic planning that helps organisations make an impact with software. Impact mapping helps to create better plans

and roadmaps that ensure alignment of business and delivery, and are easily adaptable to change. Impact mapping isn't the first or the last solution in its space, but it is important because it fits nicely into several current trends in software product management and release planning, including goal-oriented requirements engineering, frequent iterative delivery, agile and lean software methods, lean startup product development cycles, and design thinking.

Who is this book for?

The primary audience of this book are senior people involved in building software products or delivering software projects, from both business and delivery sides. This includes business sponsors and those whose responsibilities include product ownership, project oversight or portfolio management, architecture, business analysis, quality improvement and assurance and delivery. My experience is mostly with iterative delivery, so I write from that perspective, and you'll get most out of these ideas if you work in a similar environment.

- Business people assigned to software projects will learn how to communicate their ideas better.
- Senior product or project sponsors will learn how to communicate their assumptions more effectively to delivery teams, how to engage delivery teams to make better strategic decisions, and how to manage their project portfolio more effectively.

- Delivery teams that are already working under the umbrella of agile or lean delivery methods, and more recently lean startup ideas, will learn how to better focus deliverables and engage business sponsors and users.
- Delivery teams moving to agile or lean delivery methods will get ideas on how to address some common issues with scaling these practices, such as creating a big picture view, splitting work into small chunks that still have business value and reporting progress more meaningfully.

Giving credit where credit is due

Impact mapping is a variant of the InUse effect mapping method, introduced by Mijo Balic and Ingrid Domingues (Ottersten), combined with impact maps for training organisations invented by Robert O. Brinkerhoff, the feature injection ideas of Chris Matts, the measurability and iterative delivery ideas of Tom Gilb. It draws heavily on their work – enough to say that all key good ideas are theirs, I've just linked them together and put them into the perspective of modern software delivery practices. You will find references to the original ideas at the end of this booklet. Most of the glue between these ideas comes from inspiring, insightful and challenging conversations I've had with Craig Larman, Tom and Mary Poppendieck, Dan North, Gordon Weir, Jeff Patton and Matthias Edinger (in no particular order).

By combining these ideas, impact mapping brings usability and speed to proven product and project management strategies, helping them fit better into modern software delivery constraints, and at the same time applying some great ideas from other industries to software delivery.

This book describes the way I use impact maps. In my previous work, I referred to the method described here as

'effect mapping', as the structure closely resembles InUse effect maps. However, the way I use the maps is significantly different from the InUse method. I follow an approach much closer to what Brinkerhoff describes in his HET (highly-effective training) method, as roadmaps and iteratively refined milestone plans. In addition, I found that a slightly modified set of questions fits better the kinds of projects that I am involved in. InUse effect maps aim to facilitate innovative product design and user experience design. As a consultant, I work with ambitious organisations to help them improve delivery practices, and they often suffer from scope creep, lack of big picture, lack of alignment of delivery teams with business objectives; they waste a lot of time and effort building the wrong software. Impact mapping is a fantastic way to reduce all that suffering.

Using the same name for both InUse effect maps and the maps as I use them caused confusion, to the point that a well-known consultant said to one of my clients, 'Gojko got effect maps completely wrong, but he's on to something'. After several conference presentations in Sweden, the home of InUse, attendees came to complain that I was presenting effect mapping wrongly. This is why in this book, and from now on, I will use the name impact maps instead of effect maps for my method. By using a different name, I hope to prevent further confusion.

Craig Larman suggested the name 'impact maps', which is similar enough to effect maps but different enough

not to cause confusion. Brinkerhoff calls his planning visualisation impact maps, which justifies the use of that name. His maps are used for managing training plans for organisations, so I hope that there is not much danger of confusion.

Disconnecting the maps I describe here from InUse effect mapping by name also allows me to focus completely on managing scope for software delivery, and use names for map elements that are more appropriate for that context.

Make an impact!

I believe that impact mapping is a game-changing method that can significantly improve the way many teams and organisations build products and deliver projects. My aim with this book is to raise awareness about this method and related ideas and stir up the community. That is why the book is intentionally short, so that you can read it quickly and keep it close as a quick reference. Instead of trying to cover everything in detail, I provide plenty of references that will enable you to dive deeper into related topics.

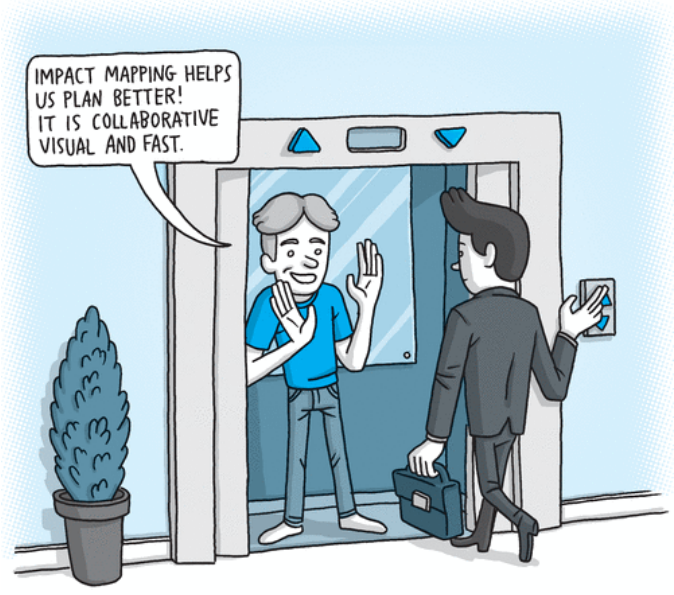
As a new method that glues together many important trends in the industry, impact mapping will hopefully evolve with the community. This is where I need your help. Try it out, see what works and what needs to be changed in your context, and engage with the community to help us evolve it. For information on how to discuss

your findings and experience with other practitioners visit impactmapping.org

Also, please consider reviewing this book on Amazon or similar online stores. The number of ratings on Amazon and reviews, even if they are one-liners, make a huge impact on a reputation of a book. Your review will help influence other people to get interested in impact mapping and spread these ideas further.

Once we collect enough experience in using this technique in different contexts, I plan to encourage others to write about this and publish additional guides. In several years, we will hopefully have enough for an implementation guide with real case studies and experience reports from different contexts. To be notified when follow-up videos, articles and books become available, register at gojko.net/impact.

Why should you care?



Impact mapping is a strategic planning technique. It prevents organisations from getting lost while building products and delivering projects, by clearly communicating assumptions, helping teams align their activities with overall business objectives and make better roadmap decisions.

Our products and projects do not work in a vacuum. They

have an interdependent, dynamic relationship with people, other projects, the organisation and the wider community around them. Yet currently popular planning methods either expect the world to stand still while we deliver or give up on creating any kind of long-term big-picture view, leaving a huge communication gap between business sponsors and delivery teams. Impact maps visualise the dynamic relationship between delivery plans and the world around them, capturing the most important assumptions as well as delivery scope. They help us adapt plans effectively and react to change, while still providing a good road map for delivery teams and a big-picture view for business sponsors.

Impact mapping helps to reduce waste by preventing scope creep and over-engineered solutions. It provides focus for delivery by putting deliverables in the context of impacts they are supposed to achieve. It enhances collaboration by creating a shared big-picture view for technical and business people.

Similar approaches to impact mapping are used in many proven management styles, including US and UK military Mission Command and workplace empowerment in commercial organisations, as well as several goal-oriented requirements engineering methods for software. Impact mapping has several unique advantages over similar methods:

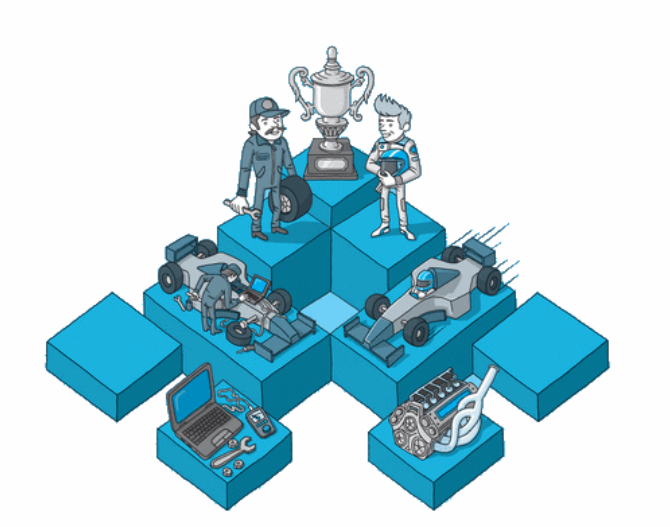
- It is based on a method invented by an interaction design agency and similar to a team-building

method, which means that it facilitates collaboration and interaction. It is significantly less bureaucratic and much easier to apply than many alternatives. It also facilitates the participation of groups of people from different backgrounds, including technical delivery experts and business users, helping organisations use the wisdom of crowds.

- It visualises assumptions. Alternative models mostly do not communicate assumptions clearly. Impact mapping does, and because of that it helps teams to make better decisions in rapidly changing environments such as IT. The visual nature of this method also facilitates effective meetings and supports big-picture thinking, which provides organisational alignment.
- It is fast. One of my clients recently said that it would take them months to achieve what we did in just two days. Because of this, it fits nicely with iterative delivery models that are now becoming mainstream in software.

In essence, you should care about impact mapping because it can help you build products and deliver projects that make an impact, not just ship software.

What is an impact map?



An impact map is a visualisation of scope and underlying assumptions, created collaboratively by senior techni-

cal and business people. It is a mind-map grown during a discussion facilitated by answering the following four questions:

1. Why?
2. Who?
3. How?
4. What?

Why?

The centre of an impact map answers the most important question: *Why are we doing this?* This is the *goal* we are trying to achieve.

It might sound like common sense to know this upfront, but from my experience very few people working on delivery know the actual expected business objectives. These are sometimes drafted in a vision document, but more frequently exist only at the back of senior stakeholders' minds. Even when they are communicated, business goals are often defined in vague terms. Knowing why we're doing something is the key to making good decisions about cost, scope and timelines, both at the start and later when things change.

Gary Klein's research with emergency services and military personnel shows that people on the ground have to know the objectives of any activity in order to react correctly to unforeseen problems. And unforeseen problems are a fact of life in any but the most trivial software. If a product milestone or project succeeds in delivering the expected business goal, it is a success from a business perspective, even if the delivered scope ends up being different from what was originally envisaged. On the other hand, if it delivers exactly the requested scope but misses

the business goal, it is a failure. This is true regardless of the fact that delivery teams can blame customers for not knowing what they want.

By having the answer to 'WHY?' in the centre, impact maps ensure that everyone knows why they are doing something. That helps teams align their activities better, identify true requirements and design better solutions.

Getting it right:

The purpose of a goal definition is to allow the delivery organisation and business sponsors to re-evaluate the plan as new information becomes available. For this reason, good goals tend to be SMART: Specific, Measurable, Action-oriented, Realistic and Timely.

Goals should not be about building products or delivering project scope. They should explain why such a thing would be useful.

Goals should present the problem to be solved, not the solution. Avoid design constraints in a goal definition.

Don't worry about nailing it down to a single number. Chris Matts suggests defining a model for business value first, and then defining goals as increments of business value, explaining how the situation should change in the future. This is particularly effective when you have a set of key performance indicators for product performance.

For commercial products and organisations, try to define goals that have an obvious link to money.

Examples:

- Starting to trade in Brazil by March next year
- Increasing user conversion by 20% in three months

Who?

The first level of an impact map provides answers to the following questions:_ Who can produce the desired effect? Who can obstruct it? Who are the consumers or users of our product? Who will be impacted by it?_ These are the *actors* who can influence the outcome.

Gerald Weinberg defined quality as 'value delivered to some person'. To deliver high-quality results, we first have to understand who these people are, and what kind of value they are looking for from our products or project outcomes. In addition to those directly getting value out of our software, we also have to consider a host of others who can make decisions that influence the success of a product milestone or the outcome of a project. Software does not work in a vacuum and it rarely controls all the actors who are involved with it. People have their own needs, goals and preferences, which all come into play if we truly care about achieving a business goal instead of just delivering software.

Yet most requirements models completely ignore this – they focus on what the software should do and not who will benefit from it and who will be worse off when it is delivered. Then somewhere mid-work a new actor appears from nowhere and everything changes fundamentally, or

someone with sufficient decision-making influence just stops the delivery in its tracks.

Impact maps make us think about all these decision-makers, user groups and customer segments. By mapping out different actors, we can prioritise work better – for example focusing on satisfying the most important actors first.

Getting it right

Important actors are those who can significantly influence the success of a project or product milestone, including end-users and internal or external decision-makers. Alistair Cockburn advises looking for three types of actors:

1. Primary actors, whose goals are fulfilled, for example players of a gaming system
2. Secondary actors, who provide services, for example the fraud prevention team
3. Off-stage actors, who have an interest in the behaviours, but are not directly benefiting or providing a service, for example regulators or senior decision-makers

Be specific. Avoid generic terms such as 'users' – different categories of users might have different needs, and not all users of a system might be important to consider for a particular project. Try to define actors in this order:

specific individual, user persona, role or job title, group or department.

Examples:

- Mike Smith from the marketing department
- Under-18 users with a mobile device at a concert
- Apple ITunes store approvers

How?

The second level of an impact map sets the actors in the perspective of our business goal. It answers the following questions: *How should our actors' behaviour change? How can they help us to achieve the goal? How can they obstruct or prevent us from succeeding?* These are the *impacts* that we're trying to create.

Anthony Ulwick wrote that a key to successful delivery is to understand what jobs customers want to get done instead of their ideas about a product or service. This helps delivery organisations investigate different technical options and explore solutions to produce good results. It also helps to focus delivery on supporting users in getting the job done instead of just delivering features. Robert Brinkerhoff suggests something similar, but focusing on desired changes in jobs instead of just jobs that users are trying to achieve.

By listing impacts on the second level of a map, we consider the desired changes in the behaviour of actors. This leads to better plans and helps with prioritisation. Different actors could help us or obstruct us in many ways on our route to achieving the key business objectives. Some of the impacts will be competing, some conflicting, some complementary. We do not necessarily have to support all of them, but

without considering delivery scope in the context of these activities it is very difficult to prioritise and compare deliverables. The hierarchical nature of the map clearly shows who creates an impact and how that contributes to the goal. This clear visualisation allows us to decide which impacts best contribute to the goal and identify the risks; this helps immensely with prioritisation.

Getting it right:

Don't list everything an actor might want to achieve. List only the impacts that really help move you in the right direction towards the central goal.

Impacts are not product features. Avoid listing software ideas here, focus on business activities.

Ideally show a change in actor behaviour, not just the behaviour. Show how the activity is different from what is currently possible. So instead of just 'selling tickets', say 'selling tickets five times faster'.

Consider negative or hindering impacts as well as positive ones.

Important actors can often help or hinder the outcome in many different ways. Once you discover the first impact of an actor, think about what else they could do.

Examples:

- Inviting more friends
- Purchasing tickets without calling the call centre
- Selling tickets faster

What?

Once we have the first three questions answered, we can talk about scope. The third level of an impact map answers the following question: *What can we do, as an organisation or a delivery team, to support the required impacts?* These are the *deliverables*, software features and organisational activities.

Delivery plans and requirements documents are often shopping lists of features, without any context that explains why such things are important. Without a clear mapping of deliverables to business objectives, and a justification of that mapping through impacts that need to be supported, it is incredibly difficult to argue about making or not making an investment in certain items. In larger organisations with many stakeholders or product sponsors, this leads to huge scope creep as everyone's pet features and ideas are bundled in. No wonder such plans often fail.

An impact map puts all the deliverables in the context of the impacts that they are supposed to support. This helps with breaking deliverables down into independent chunks that provide clear business value, and helps us launch something valuable sooner. A clear hierarchy allows us to group related deliverables, compare them and avoid overinvesting in less important actors or impacts. It also helps

us to throw out deliverables that do not really contribute to any important impact for a particular goal. Finally, by connecting deliverables to impacts and goals, a map shows the chain of reasoning that led to a feature suggestion, visualising the assumptions of stakeholders. This allows us to scrutinise those decisions better and re-evaluate them as new information becomes available through delivery.

Getting it right:

This is the least important level of an impact map. Don't try to make it complete from the start. Refine it iteratively as you deliver.

Treat deliverables as options, don't take it for granted that everything listed here will actually be delivered.

Don't go into a lot of detail early on, there will be time for that later. List only high-level deliverables. You can break down high-level features into lower-level scope items, such as user stories, spine stories, basic or extension use cases later. These items can become fourth, fifth or sixth level map branches.

Even on software projects, there are often ways of supporting a business activity without building software – sometimes it is cheaper to pay for advertising to recruit new players than spend months rebuilding a system. Consider anything that helps to achieve an impact.

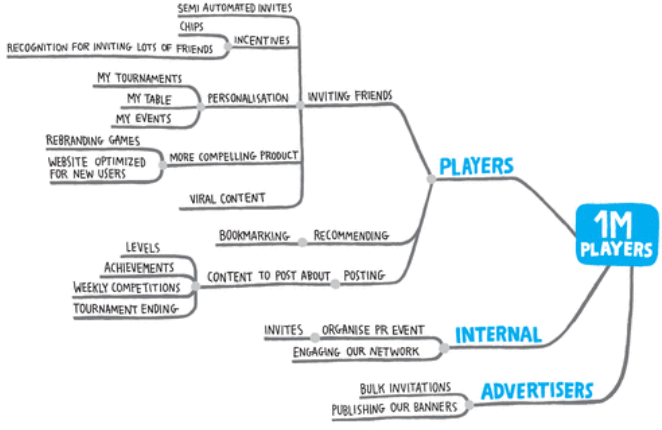
Examples

- On-line ticket sales
- Mobile home page with purchase form
- Optimise call centre sales scripts
- Sign re-selling contracts

Never aim to implement the whole map.

Instead, find the shortest path through the map to the goal!

Example: Online gaming



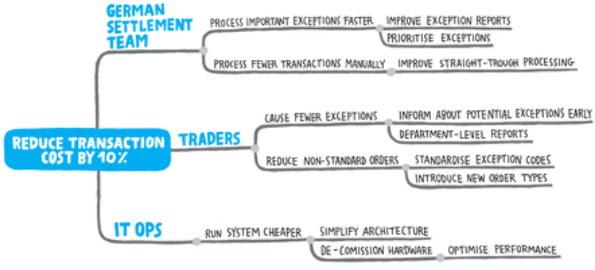
The map illustrates a milestone of an online gaming platform development. The key business goal for this milestone is to increase the number of active players to 1 million.

Players are important actors. They can help by recommending games to their friends, posting about the games on Facebook or inviting friends directly. Grouping those impacts together shows that they all contribute to the same goal, so we do not necessarily have to support all of them.

Another group of important actors are advertisers. They can also bring players by publishing adverts.

There are several deliverables that can impact player behaviour. For example, semi-automated invites can help make invitations more straightforward, a more personalised experience will make it more likely that players will invite friends, and reaching new levels or achievements should motivate players to post about the games on social networks.

Example: Financial transaction processing



The map illustrates a milestone of a transaction processing system. The key goal is to cut the costs of processing by 10%.

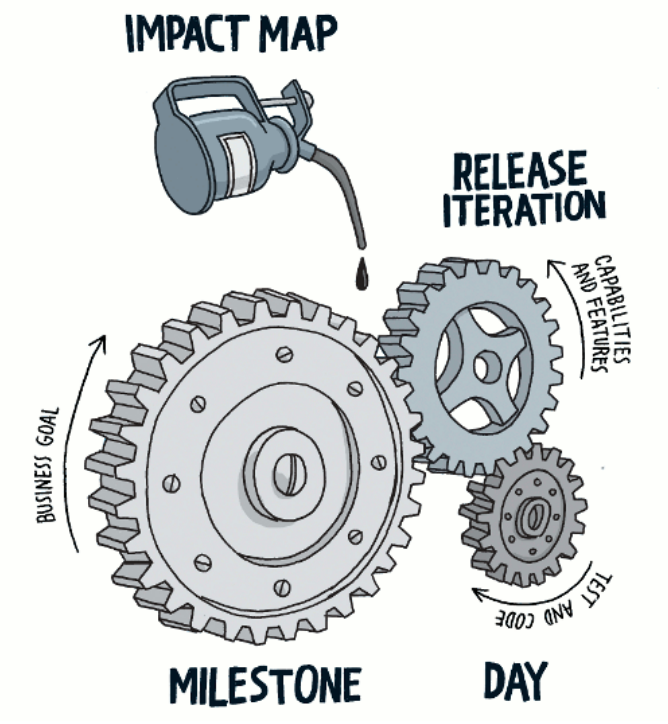
The key assumption is that cheaper IT operations would bring this saving. Some deliverables that would help with this are a simplified system architecture and the decommissioning of older systems that require expensive support. These deliverables, though, would require a lot of work and any core architectural changes are high risk.

Another option is to reduce manual work required to process transactions. The settlement team in Germany is an important actor as they could reduce the cost of trans-

actions significantly by processing the most important exceptions faster, which will reduce the work of downstream teams. Two potential deliverables could help: better exception reports and fully automated prioritisation of work. Another way to reduce cost would be to improve straight-through (fully automated) processing, directly reducing the workload of the settlement team and all downstream teams.

Traders creating orders are also important actors: they could reduce the cost of transactions by creating orders that require less manual work. There is an assumption that a significant amount of delay and unnecessary manual work is caused by the settlement team chasing after non-standard orders and free-text exception messages that are often just trader notes and not real exceptions. Perhaps we could reduce that. Standardising a list of exception codes and making traders input the codes directly instead of free-text notes is a business task, not a software delivery task, but could potentially be a quick and easy way to reduce cost.

The role of impact maps



Impact mapping is a great way to communicate assumptions, create plans and align stakeholders for iterative software delivery. Impact maps also facilitate several popular product and project management practices.

Three key roles

When everyone involved in delivery understands the objectives, expected impacts and key assumptions in the same way, products and projects benefit from better focus and less waste. This is why goal-oriented requirements engineering methods are becoming increasingly popular in software academia. Current goal-oriented requirements engineering practices mostly focus on the early requirements phase – they end where most traditional specification techniques would start. This provides a big-picture view, but it also significantly undercuts the effectiveness of such techniques in modern software delivery.

Iterative delivery methods and lean startup ideas place significant emphasis on integrating learning from delivery into refining scope, specifications and requirements. Upfront plans are inadequate because the landscape changes too frequently. But iterative delivery plans often lack a big picture.

Impact maps bridge the two worlds: they facilitate strategic planning and thinking to create a big-picture view focused on key business objectives, but also facilitate learning through delivery and help us manage project roadmaps. They represent and organise delivery scope in a way that is easy to evolve, reprioritise, grow and shrink as necessary to react to changed market opportunities or new knowledge.

Strategic planning

Impact mapping is a great way to engage senior business and technical experts at the start of work on a product module or project milestone to create a shared understanding of scope – not from a technical but from a business perspective.

Visual meeting techniques and collaboration ensure that senior decision-makers share an understanding of key business assumptions. This helps to align everyone with the overall vision.

The structure of an impact map facilitates a good discussion, helping the entire group benefit from the wisdom of crowds. This often leads to discovering quick wins, alternatives to original suggestions that are cheaper and faster to deliver but achieve the same outcomes.

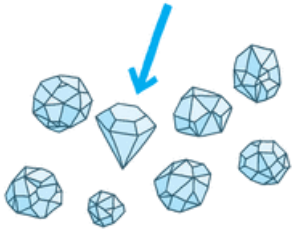
To use impact mapping for strategic planning, you need:

- strategic goals – impact mapping is not applicable to maintenance mode projects

- the participation of senior decision-makers and senior technical people

Defining quality

An impact map clearly shows the impacts that technical deliverables are supposed to produce, from a business perspective. This visualisation defines the expected quality of software at a holistic level, and ensures that everyone involved in delivery shares the same understanding.



An impact map helps the delivery organisation maintain a strong focus during delivery, and helps to prioritise and define activities related to improving or assuring quality. The role of testing becomes to prove that deliverables support desired actor behaviours, instead of comparing software features to technical expectations. If a deliverable does not support an impact, even if it works correctly from a technical perspective, it is a failure and should be treated as a problem, enhanced or removed.

To use impact maps to define quality, you need:

- agreement that the purpose of deliverables is to support changes in actor behaviour
- agreement on metrics that express stakeholder expectations for those changes

Roadmap management

An impact map communicates scope, goals and priorities, but also assumptions on two levels. The first is that a deliverable will cause a change in behaviour of an actor, produce an impact. The second is that once the impact is achieved, the relevant actor will contribute to the overall objectives. This visualisation makes impact maps a powerful tool for roadmap management.

Once a deliverable is shipped, we can measure the actual changes in actors' behaviour and the impact on the overall objective. We can then re-evaluate our strategy and decide whether to continue working on the same part of the map or move on to something else.

To use impact maps for roadmap management, you need:

- agreement that your aim is to achieve a business goal, not deliver pre-set scope
- frequent, iterative releases to measure progress
- agreement on metrics that express stakeholder expectations for the central business goal

Impact maps solve common problems

One of the unique features of impact maps, compared to alternative planning methods, is that they allow us to quickly solve common planning and delivery problems.

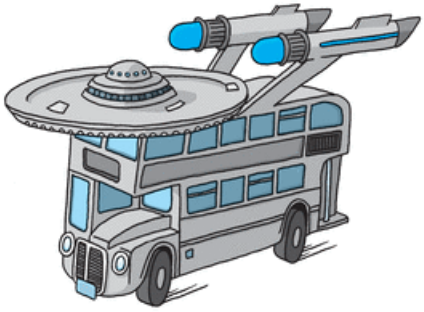
Scope creep

Because there is a clear mapping from deliverables to goals, we can measure when the main objective is reached and stop working on it. An impact map demonstrates that no further impacts are needed to achieve our goal. Similarly, maps provide us with a clear view of all the planned software features that support an impact. Once the impact

is achieved, we can stop working on the other ideas in that area of the map and move on to something else.

Wrong solutions

Because an impact map puts software deliverables in the context of business goals, it is trivially easy to spot solutions looking for a problem, or those that would contribute to a different business objective.



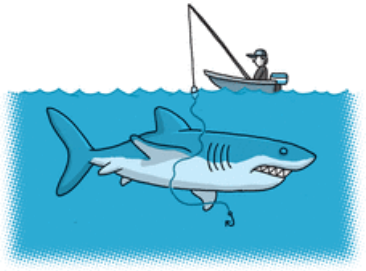
By visually grouping together different deliverables that could contribute to the same impact, maps allow us to spot over-complicated solutions and prevent over-engineering. They facilitate thinking about alternatives, helping delivery teams come up with simpler, cheaper and faster alternative deliverables that achieve the desired effect. Because of this, impact maps make it is easy to argue that more complex solutions should be discarded or at least postponed until simpler suggestions are proven wrong.

Pet features

Impact maps allow us to quickly spot suggested features that do not support any of the desired activities, or do not contribute to the overall goal. They just won't fit anywhere on the map visually. This helps us discard or postpone such pet features.

Wrong assumptions

Even with the best insight, if market opportunities change, it is impossible to know that a solution is no longer relevant unless underlying assumptions are clearly communicated. Impact maps clearly show assumptions so that we can track and validate them.



Ad-hoc prioritisation

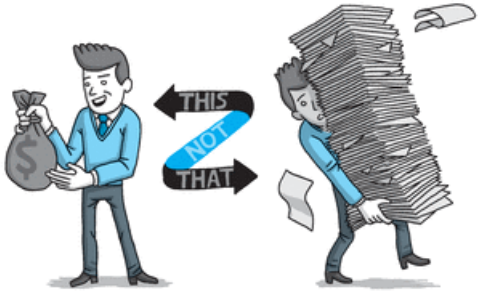
Without a clear understanding of the business context, it is difficult to know that we're really working on the most important things.

Impact maps put features in the context of behaviour impacts, allowing stakeholders to relate better to benefits they would get from features and make much better prioritisation decisions. We can make decisions based on which

impacts to support first, or which actors to satisfy, bundling together all the necessary deliverables to achieve this goal. This significantly reduces time to market.

In larger organisations, impact maps also help to communicate the big picture to many stakeholders that need to coordinate their decisions. This significantly helps align everyone's expectations, and make trade-offs between the priorities of different departments.

Delivering business goals, not just features



Based on a seven year long research of companies across the European Union, John McManus and Trevor Wood-Harper list the following as the top five key reasons why IT projects fail:

- Business reasons
- Business strategy superseded
- Business processes change (poor alignment)
- Poor requirements management
- Business benefits not clearly communicated or over-stated

Clearly, current practices in the IT industry for managing project goals and communicating requirements totally fail to deliver business outcomes. This is not a problem only in IT. Studies of military commanders and tank platoon leaders in the US Army show that giving answers to 'what' and 'how' does not prepare individual teams for reacting to unforeseen problems. Yet this is exactly what most software requirements models communicate. Explaining 'why' and 'watch out for' is much more important in a fast-moving landscape such as software delivery.

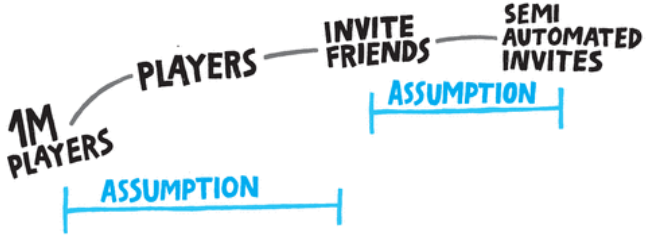
This is what goal-oriented requirements practices try to achieve. The idea with goal-oriented requirements practices is to focus projects on supporting usage goals and behaviour impacts, instead of pre-set features or tasks. These ideas have been around for a while and gained popularity particularly in the early 2000s, with some renewed recent interest in academia, especially with the I* model. In my opinion, the I* model is just too complex for wide industry adoption.

Scott Berkun used a simple mapping between goals, features and requirements to prevent scope creep while managing the Excel development program at Microsoft. He got the stakeholders to agree on a limit on the number of business goals for a release. Then he eliminated superfluous scope by insisting that each feature references a requirement it fulfils, and that each requirement references the goal it supports. Berkun wrote that such mapping allowed his teams to decide which features should or should not be

part of a delivery, which ones should be reprioritised when business goals changed and which were critical for success.

Impact maps facilitate this process, taking it even further. Instead of just connecting features to goals at the start of a product milestone or a project, impact maps allow us to maintain a dynamic roadmap that changes with our learning, keeping the goal in mind and making features and scope secondary to it. We visualise assumptions, which allows us to change direction once these assumptions are proven or discarded.

For example, the online gaming system example from Part I of this book shows that the purpose of semi-automated invites is to support players inviting their friends. We assume that delivering that feature will change players' behaviour. Once the feature is delivered, we can track if the assumption was true or not. If players invite friends, we can keep investing in that branch. If not, we should stop, analyse the situation, and try to do something else about invitations.



The map also shows a higher level assumption. If players send out invitations, we assume that their friends will actually come to play our games. After each deliverable is shipped in the invitation area of the map, we can measure if new players are responding to invitations at expected rates. If yes, we should continue to expand this area. If not, building more ways for people to send out invitations would be wrong. We should re-think our strategy, find a way to improve invitation results or perhaps do something else.

Finally, by mapping deliverables to a business goal, impact maps strengthen the case for limiting the number of business goals and impacts we work on at any given time, in line with the idea of limited work-in-progress from lean development methods.

Further reading:

- Sources of Power [Klein99]
- A study in project failure [McManus08]
- Goal-Oriented Requirements Engineering [Lapouchnian05]
- Social Modeling for Requirements Engineering [Yu11]
- Competitive Engineering [Gilb05]

Setting measurable objectives

Just saying that we focus on achieving a business goal isn't enough – we have to know that we're focusing on the right goals. Many projects fail not because they don't have a business goal, but because the goal is vague, conflicts with other objectives or is not deliverable by the group of people tasked to implement it. Scott Berkun uses the acronym SMART to identify good business goals: Specific, Measurable, Action-oriented, Realistic and Timely.

Measurability is the key element here – if a goal is measurable, the other elements are easier to pin down. For example, 'more users' is not a good goal. The solution that

delivers 50% more users over a year is probably completely different from the one that increases user numbers by 200% over two months. Once we have an agreement about how much and over what time, it becomes much easier to identify whether we can deliver it. In *Competitive Engineering*, Tom Gilb shows how making objectives measurable helps to discover if we have identified the right objectives, and to focus projects on the right objectives. Thinking about measurability has helped me to stop several death-march projects before they even started, align everyone's expectations and define precise and deliverable outcomes.

Impact maps fit in well with the SMART approach, because they help us apply the measurability test not only to the central business goal, but also to the impacts that we aim to support.

Defining SMART goals and impacts complements the big-picture approach of impact mapping. Both techniques help us to discover unrealistic expectations more easily, discard solutions that won't get us to the target and identify the simpler solution.

Avoiding vanity metrics

In *How to Measure Anything*, Douglas Hubbard warns about a common psychological tendency to track what is easy to measure instead of what can inform good decisions. Hubbard says that information is valuable if it reduces uncertainty about decisions, and that the "value of measuring

a variable is often inversely proportional to the attention it gets". Knowing that we need to decide whether or not to continue investing in an area of a map, we can design measurements that would inform this decision.

In *Lean Startup*, Eric Ries warns against vanity metrics, whose sole purpose is to make people feel good, as a key problem affecting the success of software delivery. He advises focusing on metrics that can lead to actions. Impact maps can help establish good actionable metrics, because they make us think about desired impacts on actor behaviour and how we can observe them.

Two levels of metrics

Metrics are very sharp tools. They are great at helping us focus, but can hurt badly if we focus on the wrong things. A common problem with measurements is that they can change from being the means to control the success of a project to an end in itself. In *Seeing like a state*, James C. Scott warns about the conflict between two levels of metrics: the ones that provide understanding and control to people outside the process being measured, and the ones that are important for the people inside, affected by the process. Citing examples from building large cities to scientific forestry, Scott shows that imposing metrics for outside control often makes a system overly complex and illegible to those inside, leading to huge failures. For example, several economic disasters in organising large-scale farms were caused by focusing on land area metrics and

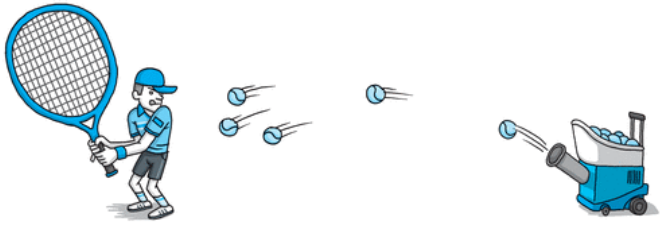
sowing quotas, dividing plots and work in a way that was easy to control and understand to government authorities. Such plans were often impossible to implement, because plots were nicely divided in rectangular shapes on paper, but had rivers and marshes in real life, and different soil types. Outside measures failed to consider things important for farmers, such as potential yields, survivability and the needs of different crops. Scott also shows how projects with strong outside metrics can fail when people involved focus on reaching quotas even when this damages the overall objectives, but achieves their private agenda.

Impact maps help us prevent disasters such as this by making us think about the people involved and the impacts that we want to create. Goal metrics do not become an end in themselves, as we take care of achieving various impacts visible on the map. We can use the metrics associated with the business goal to control the delivery and make it understandable from the outside, but we can also associate metrics with actor impacts, considering things that are important and meaningful for the people inside.

Further reading:

- Art of Project Management [Berkun05]
- How to Measure Anything [Hubbard10]
- Lean Startup [Ries11]
- Seeing like a state [Scott99]

Adaptive planning



Scott shows that big upfront plans rarely survive contact with reality because of individual variations and local knowledge, saying that 'too many unknowns in urban planning rendered any solution problematic or else required heroic assumptions'. This is why plans driven by metrics need a counter-balance: adaptive planning.

Similarly, big upfront software plans rarely survive contact with reality because of an ever-changing technical landscape and the individual goals and preferences of the people involved. We have to make heroic assumptions all the time when delivering software. Impact maps allow us to do that, because they provide the basis for feedback on those assumptions. They visualise assumptions so that we can control them, and adapt our plans accordingly.

Tim Harford's book *Adapt* is full of amazing accounts

of big plans that turned into epic disasters, from 19th century imperial Russian failures to organise mining to US military attempts to control Iraqi insurgencies and the recent meltdown of the banking sector. Harford compares those failures with successes such as the Spitfire airplane and Google's internet domination, suggesting the solution in the form of a checklist:

- Variation: seek out new ideas and try new things
- Survivability: try new things out at a scale where failure is survivable
- Selection: seek out feedback and learn from your mistakes as you go along

Harford calls this checklist the Palchinsky method, after an early 20th century Russian engineer Peter Palchinsky, who recognised that 'real-world problems are more complex than we think', because of they have a human dimension and a local dimension, and are likely to change as circumstances change. I'm sure this will sound familiar to anyone who has ever been involved in a real-world software project. As the key aspects of this solution, Harford calls for adaptive planning and opening up lines of communication from the field to the planning office.

Impact maps can help to facilitate the application of the Palchinsky method. By focusing on business objectives and desired impacts, a map helps us think better about alternative ways of delivering, stimulating variation. While

mapping impacts we break deliverables into small pieces, supporting survivability. Finally, by visualising assumptions and defining a clear path from deliverables to desired impacts and business goals, impact maps define good measurements for feedback, helping with the selection principle.

Build-measure-learn

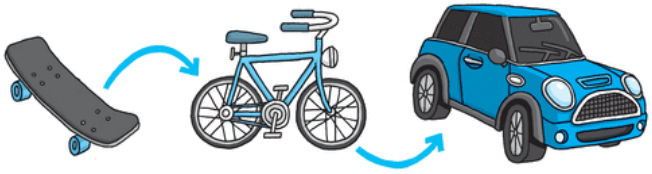
Lean startup is an innovative approach to designing and implementing products and managing companies, described by Eric Ries in his book *Lean Startup*. Two of the core ideas of lean startup are validated learning (intentionally testing assumptions) and the build-measure-learn cycle (small iterative deliveries to facilitate learning). Ries' book does not say much on how to choose which assumptions to test or how to prioritise experiments for the build-measure-learn cycle. Impact mapping complements the lean startup ideas nicely by enabling us to clearly visualise assumptions and see the big picture, so that we can decide which experiments are the most valuable to conduct and most in line with overall company goals.

For example, if we decide to run an experiment on invitations, we're aiming to validate an assumption that semi-automated invites will support players inviting their friends. The metrics we establish for this impact define how many invitations we expect, over what period etc. We build the feature, measure the results, compare them to expectations and learn from this to decide on the next feature. We apply this to the larger picture as well: improve invitations, measure the rate at which invited friends are becoming new players, learn from this and decide where to invest next.

Further reading:

- Lean Startup [Ries11]
- Adapt [Harford11]

Iterative delivery



The environment in which we deliver IT products and projects changes too frequently for detailed long-term plans and upfront design. Iterative delivery models accept this as a reality, and allow us to integrate learning into ongoing work and improve solutions during delivery. Iterative delivery became mainstream during the last decade, especially through the use of agile methods. As an industry, we got a lot better at integrating feedback to improve technical delivery, but we still have a long way to go in order to improve business decision-making based on delivery feedback.

A recent report published by Forrester Research argues that most organisations apply agile ideas only to development, leaving business outside the loop and losing out on big benefits. Many organisations adopt a process that Dave West called Water-Scrum-Fall, with a huge business decision-making part up front, an iterative technical delivery part

in the middle, and a long business approval process at the end. West's conclusion that "most firms do requirements and planning before forming a Scrum team" paints a bleak picture of agile adoption. Impact maps can help solve this problem as they make business and technical experts clarify business objectives, focus on achieving impacts and think about software scope ideas as assumptions that could be proven right or wrong.

Big picture

A common cause for the disconnect between business and delivery is that teams iteratively deliver items that are too small to make a difference from a business perspective. Projects become endless streams of small items going through the pipeline, with no big picture or big benefits. Business sponsors simply don't care about small deliverables, and they wait for everything to be finished. Even worse, they often ask for scope and cost to be fixed at the beginning of a project, pretty much defeating the point of iterative delivery. By trading the big picture for the ability to change things quickly, many teams lead their projects very much as if they were making their way along a dark tunnel – we don't know exactly where we'll go but small steps ensure we do not fall down.

Impact maps provide the big picture for both business sponsors and technical delivery teams. They remove the need for major upfront analysis. Because the process of creating

a map is quick, they fit nicely into iterative delivery models. I've used impact maps with great success to complement more established iterative requirements techniques. With a map, we can report progress on a much more meaningful level – showing which impacts we've supported or where we plan to focus our efforts next from a business perspective. Impact maps allow us to commit on impacts we plan to support, not on software features that we plan to implement, providing visibility and commitment to project sponsors while still allowing flexibility in implementation.

Better prioritisation

Many teams fail to engage business users because they ask them to make decisions on the priority of software features. Impact maps allow us to have a discussion around business activities and impacts (level 2), instead of deliverables and software features (level 3 and beyond). In my experience business users are much better engaged and make much better decisions when the discussion is at that level.

Iterative, not incremental

A common way teams fail with iterative delivery is to incrementally deliver pieces that only bring value when they come together at the end. Jeff Patton famously illustrated the difference between incremental and iterative delivery with his Mona Lisa drawing.

Impact maps break deliverables down into branches that contribute to individual impacts, and make us think about quick ways to support them, promoting truly iterative delivery.

Further reading:

- Water-Scrum-Fall [West11]
- I don't know what I want but I know how to get it [Patton08a]

User stories

User stories are de-facto standard for managing scope in agile methods today. They are intended to support teams in planning scope without a huge investment in upfront analysis and to ensure that each item brings clear business benefits.

Many organisations fail to get the big benefits of user stories by creating too many stories at the beginning, describing the entire scope in order to keep track of everything that needs to be delivered. Major upfront analysis

is reduced, but we're left with a big story list. This leads to a huge number of stories that need to be tracked and managed, which is a waste of time. Even worse, when business circumstances change, such lists require a lot of effort to reprioritise and understand. Jim Shore calls this 'story card hell'.

Impact maps prevent story card hell. The reason why organisations get into this situation is that it is hard to let go of long-term plans. Business stakeholders who worry about their needs being forgotten insist that those needs get captured somehow. Instead of writing hundreds of low-level stories, impact maps allow us to capture stakeholder needs as actor behaviour impacts. We do not really need to discuss scope for anything but the most important actor impact for the moment. Once we start working on an impact, we can grow the scope. Instead of hundreds of stories, we can work with a map and a dozen stories at a time.

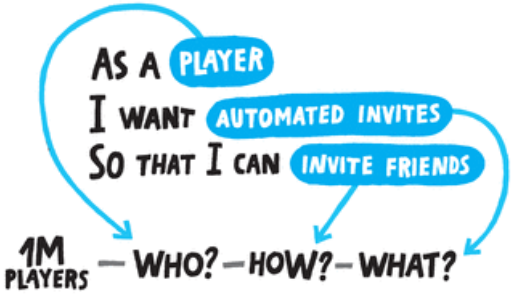
With a hierarchy of visible business assumptions and impacts, impact maps allow us to prioritise whole branches easily. If the market opportunities change and one of the assumptions no longer holds, we can easily identify all the in-scope items that should be killed.

Keeping user stories honest

Another common problem with user stories, especially when they are created in bulk, is that they are often created

to describe technical deliverables. The most popular story format lists the expected business benefit, a user group that will benefit from the story and the scope that will be delivered on a high level. The idea with this format is to ensure that we know why each story is important, so that we can plan better.

Yet when someone wants a pet feature, they often justify it by wrapping it in a user story format. Such stories often have vague, missing or circular stakeholder and business value parts. 'As a trader I want to trade because I want to trade' is a classic example, along with 'As a system I want a customer report'.



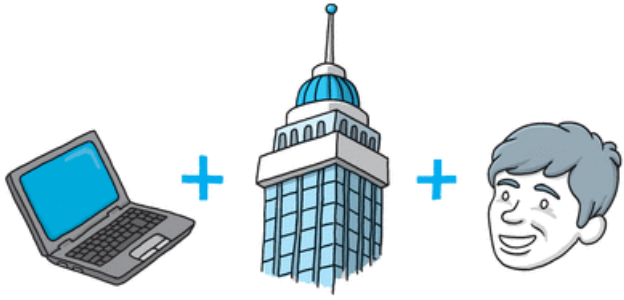
Impact maps help to keep user stories honest – each story needs to fit somewhere on the map. In order to fit it in, the stakeholder and business benefit on the story card should relate to an actor and an impact on a map. This makes people think harder about what they really want to achieve with a user story, and formulate them better. If there is no

suitable place for a story on the impact map, it probably doesn't belong in the current delivery cycle.

Further reading:

- User Stories Applied [Cohn04]
- Beyond Story Cards [Shore05]

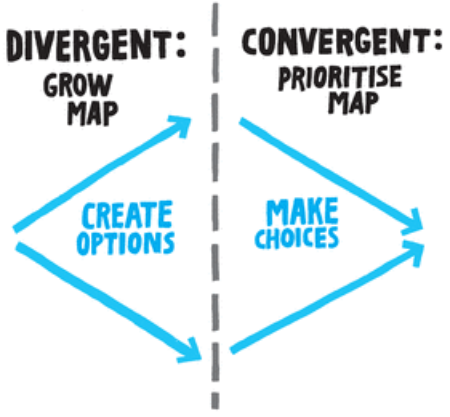
Design thinking



Design thinking is a problem-solving approach that applies creative design techniques to drive innovation and foster business growth. The movement was started in the design community by the creative agency IDEO and in the last decade crossed over from design into business management. The promise of design thinking is that organisations will build better products and innovate by applying a designer's toolkit to balance the needs of people, the possibilities of technology and the business success requirements.

Tim Brown writes that design thinking is 'a dance among four mental states': divergent and convergent thinking and analysis and synthesis. In the divergent phase, teams gener-

ate options to explore. In the convergent phase, they decide which of those options are worth pursuing further. In the analysis phase they collect information to gain insights into the chosen options, but, as Tim Brown says, 'the facts never speak for themselves'. In the synthesis phase teams extract meaningful patterns from the information to gain confidence and learn.



Impact maps can facilitate nicely the first two mental states, divergent and convergent thinking, capturing the options and allowing us to discuss them. They can also help determine what information to collect for the analysis phase.

As we explore different options, an impact map helps us organise the ideas so that we can prioritise and analyse them better. By clearly representing actors and impacts,

impact maps can inspire people to think about alternatives and better ways that support the usage goals. During the convergent phase, the hierarchical nature of an impact map allows us to choose actors and impacts on a higher level than just analysing software feature ideas would.

Impact maps can also facilitate four out of the ten key tools for design thinking listed by Jeanne Liedtka and Tim Ogilvie in *Designing for Growth*:

- Mind-mapping to generate insights from exploration activities: the visualisation provided by an impact map is a good way to capture insights and ideas and structure them for further analysis.
- Brainstorming to generate new ideas: with the structured process of asking the four key questions and looking for alternatives (explained in the following part of this book), impact mapping facilitates brainstorming.
- Testing assumptions to prove them: impact maps capture assumptions in pictures, helping us to intentionally look for insights that will prove or reject those assumptions.
- Enrolling customers to participate in designing a solution: the visual and collaborative aspects of impact mapping help to engage both delivery teams and customers or business users in defining a good solution together.

Further reading:

- Change by Design [Brown09]
- Designing for Growth [Liedtka11]

Effective meetings

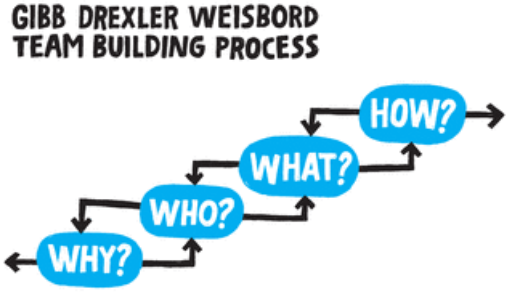
Visual facilitation makes meetings more effective, according to David Sibbet, because of three key factors:

- Participation significantly increases as people interact over the graphical presentation of ideas
- Big-picture thinking allows groups to compare ideas, discover patterns and find new ideas more easily
- Group memory improves because the meeting is recorded in pictures, and follow-through improves.

Sibbet claims that visualisations make groups smarter, writing 'visualization is worth 80 IQ points' because it releases energy, intelligence and creativity in the room.

Impact maps are visual. Senior technical and business stakeholders work together to draw maps on whiteboards, benefiting from visual meeting factors and getting those extra 80 IQ points with increased energy and creativity. This makes planning meetings much more effective. Many of my clients have been amazed how much we achieved in one or two days of impact mapping, comparing that to typical death-by-PowerPoint strategy meetings. Perhaps it is an extreme case, but one client said that they would

typically get to the same result in six months of planning meetings.

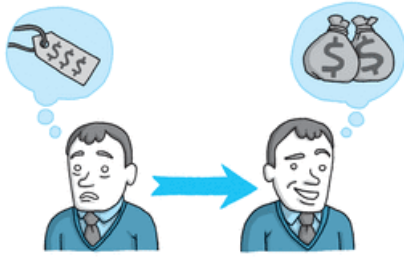


Another interesting aspect of impact mapping, in the context of productive meetings, is the similarity of the questions we use to build maps and the structure of the Gibb-Drexler-Weisbord team-building process. Jack Gibb, Marvin Weisbord and Alan Drexler formulated a team-building process in the 1960s, based on research into group processes and organisational development. They structured an engagement model around four key questions: 'Why are we here?', 'Who are you?', 'What are we doing?' and 'How are we going to do it?'. Their model helps with alignment of teams, and directly corresponds to the discussions used to create an impact map. In practice, this means that the structure of an impact map facilitates a good discussion and brings separate roles together into a single group with a common goal.

Further reading:

- Visual Meetings [Sibbet10]

Switch from cost to investment



Without a clear big-picture view of what is being delivered and why, many organisations treat IT as a cost. Delivery reports are based on time and effort spent to produce something, caring about cost more than about outcomes and benefits achieved by delivery. Impact maps visualise deliverables and assumptions and link them to business goals. This helps us justify every feature we build or test. But we can take this even further.

As with any other planning method, the first question decision-makers ask is: 'How much will this cost?' and the second one is: 'When will it be delivered?'. This is where most attempts to deliver solutions iteratively stumble, and

teams fall back on building long-term detailed plans rather than iterative delivery, losing out on the big benefits of iterative learning.

With a clear focus on business goals and good metrics to tell us when we've delivered, we can respond to questions about costs and times by asking 'How valuable is this? How much do you want to invest? And when do you want it?'. We can then add that information as key measurements to the milestone. Many business sponsors and clients already have in mind the price they are willing to pay and the deadline they require before asking for estimates, so the whole game is pretty much fruitless anyway.

Of course, just saying that we'll spend all the money and time available isn't adequate, which is why they try to negotiate in the first place. What we can do is visualise assumptions and risks and ask for incremental investment.

Good goals should be translatable to money, so we can decide what the appropriate return on investment is. This discussion alone helped me kill several doomed projects before they even started. If there seems to be enough money and time to try to deliver, we can take a small percentage of the budget and try to prove the key assumptions. If this fails, again we can stop the project before spending all the money. If the key assumptions turn out to be true, we can keep investing incrementally until we achieve the goal, and stop if we reach it before spending all the budget. Instead of tracking effort and the number of tasks implemented, we can provide stakeholders with much more meaningful

progress reports, talking about achieved impacts and key goal measurements.

With this approach, we can stop wasting time on the non-problem of estimating costs and time. Instead of adversarial negotiations on cost and effort already spent, we can have much more productive discussions on when things are needed and how much it makes sense to invest in developing them.

Creating an impact map

To get the most out of impact mapping, work in groups with senior technical and business people. When a group of people approaches the same problem from different perspectives, we leverage the wisdom of crowds, and obtain good results much sooner than if a single person creates the map. The group also gets the additional benefit of shared understanding of goals and assumptions, enabling them to plan and prioritise more effectively.

A good, measurable goal is required as input for impact mapping. It is often useful to schedule two sessions: the first to define the expected business goals and measurements, and the second to create a map.

Preparation

1. Discover real goals
2. Define good measurements
3. Plan your first milestone

Mapping

1. Draw the map skeleton
2. Find alternatives
3. Identify key priorities
4. Earn or learn

Preparation step 1: Discover real goals

Something in human psychology makes us come up with features as solutions. Most of my past projects started with business sponsors asking for features, a kind of shopping list. Business objectives often exist only in the back of some senior sponsor's head, but they aren't clearly expressed or known to anyone else. This sets up software projects for failure from the start for two reasons:

- Senior business sponsors are usually not technology experts, so the solutions they propose might not be optimal (cheapest, fastest to implement, least risky or whatever makes solutions optimal in your context). When the business goals are not communicated clearly, technology experts cannot suggest a better solution if it exists.
- Even when the initial solutions are the right thing to do now, the technology landscape changes so fast that the underlying assumptions might stop being valid during the implementation of the solution. When business objectives are not communicated clearly, it is impossible to spot that the situation has changed and that a different solution is needed.

The first step in defining a good roadmap is to make sure that we have a clear mission statement that everyone agrees on. The best way to define this mission statement is by expressing the expected business goals.

Get the group to identify possible goals and list them on a whiteboard, so that you can argue them and compare them. Make sure to get agreement on the number of objectives deliverable by the project or milestone. Try to get the group to agree on limiting this to one objective per milestone if possible.

If you can't find any objectives, start with suggested features and ask whose behaviour will this change and in what way. Then ask why those behaviour changes are important. Alternatively, 'why is the feature important' or 'how would it be useful'. Keep asking the questions until you get to the money. This is similar to the lean 5-Whys technique. Chris Matts says that you should get to something that clearly shows how to save money, earn money or protect money.


THE NEXT REQUIREMENT IS ADDING ACHIEVEMENTS TO ALL OUR GAMES.
OK, SO THAT I UNDERSTAND BETTER, HOW WOULD ACHIEVEMENTS BE USEFUL TO YOU?
PLAYERS WOULD POST ABOUT THEM ON FACEBOOK..
OK! WHY IS THAT IMPORTANT?
OTHERS WOULD READ THOSE POSTS.
AND?
THEY MIGHT COME TO BUY OUR GAMES.
SO, YOU ACTUALLY WANT TO SELL MORE GAMES!

Preparation step 2: Define good measurements

Clearly defined goals are necessary when you draw an impact map. I find that even when teams have identified business objectives (such as 'increase number of players') discussing how to measure them opens up a good conversation on viability and priorities. After you review the shopping list (scope documents, feature lists, suggested stories etc), and identify potential business goals, make them more concrete by defining key measurements.

Tom Gilb has good advice on how to establish precise measurements for goals in his book *Competitive Engineering*. A simplified version of his list that worked well for me has the following five items:

- what we'll measure (Gilb calls it 'scale')
- how we'll measure it ('meter')
- what the situation is like now ('benchmark')
- the minimum acceptable value, the break-even point
- for investment ('constraint')
- the desired value ('target')

Discussing and contrasting these measurements for different impacts leads to much more focused delivery. We can compare the goals of reducing IT costs and increasing the number of players, and argue that a slight increase in IT costs would be okay if we increase player numbers significantly. Challenge everything, in particular targets and constraints. Make sure that the targets are realistic and deliverable within the allocated time frame.

I recently facilitated a workshop where the initial list had roughly 20 goals. By discussing measurements and targets, we dropped 17 of those 20 goals – they weren't that important.

Don't worry if you do not have all the numbers at this point, assign actions for people to get the numbers before the next meeting. This is why it's good to plan for two meetings, with enough time between them for follow-up activities. Beware of measuring what is easy to track instead of what is valuable. Douglas Hubbard advises that measurements of high value have a lot of uncertainty with a high risk of being wrong.

	MORE PLAYERS IN 6 MONTHS	OPERATIONAL COSTS	PLAYER RETENTION
SCALE	# MONTHLY ACTIVE PLAYERS	HOSTING COSTS + OPS SALARIES	% PLAYERS COMING BACK ONE WEEK AFTER SIGN-UP
METER	GAME DATABASE	FINANCIAL ACCOUNTS	GAME DATABASE
BENCHMARK	350 000	$ 50, 000	32, 00%
CONSTRAINT	800 000	$ 100, 000	32, 00%
TARGET	1000 000	$ 30, 000	70, 00%

Experiment with names

Do not get stuck on the naming of these measurement points. People working in different domains relate better to different names. For example, 'break-even' is much easier to relate to than 'minimum' or 'constraint' for financial performance (eg revenue, cost)

Preparation step 3: Plan your first milestone

Once you have a list of goals and measurements, scrutinise it. Check if everything is really part of the current milestone or if you can postpone some items. There is typically one key driving objective in a shopping list – get business people to agree on it. I often use dot-voting, where everyone in the room gets a number of dots to put next to objectives. Sometimes using virtual cash can make things more precise. Create virtual cash out of post-its (eg by writing £100 on them) and distribute money to key business people in the room, so that everyone gets fewer post-its than there are goals. Another option is to distribute Monopoly or play money, available at most toy stores. Invite business sponsors to invest money in business goals. The number of dots or the virtual cash budget gives you a list of priorities.

Teams that are capable of delivering iteratively should think about working on one goal at a time. Five consecutive milestones where each fulfils one business objective are generally much better than one milestone that fulfils five partially, because the landscape might change after you fulfil a key objective. For example, you can split a milestone with two key objectives, to increase the number of players

from 300K to 1M and to reduce IT costs by 40%, into two milestones. One is to increase the number of players without negatively impacting IT costs, the other is to reduce IT costs without negatively impacting the number of players. The second objective becomes a measurement criterion for the first when the milestones are split.

If it would take too long or would be too risky to wait for a 700K increase in the number of users before reducing IT costs, renegotiate targets. The first milestone could be to increase the number of users by 100K, then the second could be to reduce IT costs, then you could return to targeting a further increase in the number of players. Once this first milestone is reached, you can come back to these measurements and define the next one.

Milestone I:

More players, no negative impact on retention, 100% increase in IT costs allowed if necessary

	MORE PLAYERS IN 6 MONTHS	OPERATIONAL COSTS	PLAYER RETENTION
SCALE	# MONTHLY ACTIVE PLAYERS	HOSTING COSTS + OPS SALARIES	% PLAYERS COMING BACK ONE WEEK AFTER SIGN-UP
METER	GAME DATABASE	FINANCIAL ACCOUNTS	GAME DATABASE
BENCHMARK	350 000	$ 50, 000	32, 00%
CONSTRAINT	800 000	$ 100, 000	32, 00%
TARGET	1000 000	$ 50, 000	32, 00%

Milestone II:

No damage to number of players or retention, reduce IT costs

	MORE PLAYERS IN 6 MONTHS	OPERATIONAL COSTS	PLAYER RETENTION
SCALE	# MONTHLY ACTIVE PLAYERS	HOSTING COSTS + OPS SALARIES	% PLAYERS COMING BACK ONE WEEK AFTER SIGN-UP
METER	GAME DATABASE	FINANCIAL ACCOUNTS	GAME DATABASE
BENCHMARK			
CONSTRAINT	1000 000	$ 50, 000	32, 00%
TARGET	1000 000	$ 30, 000	32, 00%

Ask: 'If we achieve the key targets for metrics with a completely different scope than planned, have we succeeded?'

If the answer is 'No', go back to start: You don't the right metrics.

Mapping step 1: Draw the map skeleton

The first step in drawing the map is to create a skeleton. Put the first milestone in the centre of the map, and connect some key high-level deliverables to it. People in the room most likely already have some ideas about the scope they would like to have delivered, and identifying the assumptions underlying those ideas is crucial to get everyone on the same page. The initial set of actors, impacts and deliverables will help to facilitate ideation and inspire people to look for alternatives.

If you're working with a group that already has requirements dictated in the form of a shopping list, pick a few crucial items from the list and map them out. The shopping lists I worked with were often the result of months of planning and approvals, so they had good ideas in them. There was no point in reinventing the wheel. On the other hand, many of the ideas in the shopping lists were based on wrong assumptions, and impact mapping helped uncover such cases quickly. Reverse-engineering a bit of the shopping list has a positive psychological effect even if the original list has no good ideas. The people who provided the list probably have a lot of emotional attachment to their ideas, and by putting them into the map skeleton we show

that we're valuing their opinions.

Don't reverse-engineer the entire shopping list – just list some key features to facilitate further discussion. Make sure to define precisely the actors and impacts, and double-check that the connections still makes sense:

- Is it realistic that the feature will contribute to the impact?
- Is the impact valid for the actor?
- Will the impact really contribute to achieve the goal?

Yes, it sounds obvious, but you'd be surprised by the number of times people answer 'no' to at least one of those questions after the initial attempt. Try to form sentences in the structure 'Someone can help us achieve our objective by doing something differently'. Watch out for overly generic categories, such as 'people' or 'users' and try to make it more qualified - for example 'mobile players'.

It's a good idea to split a large group into smaller groups and merge after 20 minutes, comparing the results. This will get you to a good skeleton sooner.


WHAT ARE WE DOING?
WHY ARE WE DOING IT?
1M PLAYERS
PLAYERS
POST ABOUT OUR GAMES
ACHIEVEMENTS
WHO WILL HELP US?
HOW WILL THEY HELP?

Mapping step 2: Find alternatives

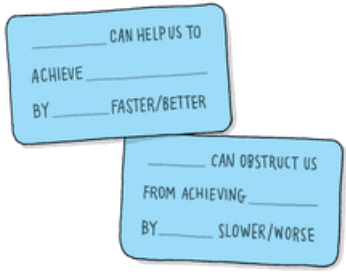
Now that you've established the key objectives and there is some structure in the map to discuss, try to define as many alternatives as you can in a time-boxed period, limiting the discussion to the actors and behaviour impacts. This corresponds to the divergent phase of design thinking. Diverge and merge. Get groups to work independently and review results every 20 minutes. The goal of this step is to try to find a better or a cheaper solution, a shorter journey to the key objectives. Don't criticise any ideas, just throw them on there. Use the existing skeleton map structure as an inspiration and ask the following questions:

- What else could those guys do for us?
- Who else can help? How?
- Who can obstruct us?

Another good trick to facilitate ideation is to give people a sheet of paper printed with a large grid. Each person has to put one idea into the first box in the grid, then pass the sheet to the person next to them. They will get a similar sheet from the person on the other side. Then they have to fill in the next box in the grid without repeating an idea

that was previously written on that piece of paper. This facilitates two things: previous ideas give people inspiration, but asking for new ideas makes people think harder. To really push things to the limit here, you can try one of the collaborative games presented in the books *Innovation Games* [Hohmann06] and *Gamestorming* [Gray10].

I sometimes pre-print cards with sentences where groups can fill in the blanks to facilitate this step.



Mapping step 3: Identify key priorities

Now comes the convergent thinking phase. With plenty of options, we need to make some choices. I try to keep this phase informal, as a natural answer often arises from a discussion. Start by asking the following questions:

- Look at possible obstructions: what are the crucial things that can stop us before we even start?
- Is there a high-value low-hanging-fruit impact somewhere?
- What are the key assumptions to test?

If this doesn't lead to a natural answer to the question of where to start, you can use dot-voting or virtual cash to get the group to select the first few items to try. Try using two colours for dot-voting: red for critical and green for low-hanging fruit. If you read the three questions above carefully, you'll see that they all revolve around business activities and impacts, not deliverables or software scope. Ask business sponsors to prioritise impacts, not deliverables (map level 3 and beyond). From my experience business users think more clearly about business activities and impacts than software features, so they can prioritise

them better as well. Another option, probably more useful for product design than project management, is to select an actor that should be satisfied first.

More structure

If you want to put more structure into this conversation, investigate the Kano and purpose-alignment models. The Kano model [Cohn06] provides a questionnaire to categorise features into mandatory (must-have), linear (more is better) and exciters (small amount can dramatically increase satisfaction). The purpose alignment model [Pixton09] breaks features down into categories of market-differentiating, parity (should be good enough), partner (non-mission-critical, buy somewhere else) and who cares (don't do) features.

Mapping step 4: Earn or learn

Having identified the actors and impacts, we can start growing the third level of the map: deliverables. In an ideal world, everything we do should directly contribute to the goal in the centre of the map. This ideal situation assumes that we'll always make the right decisions, which is an illusion. Sometimes we just don't know up front if things will turn out the way we want.

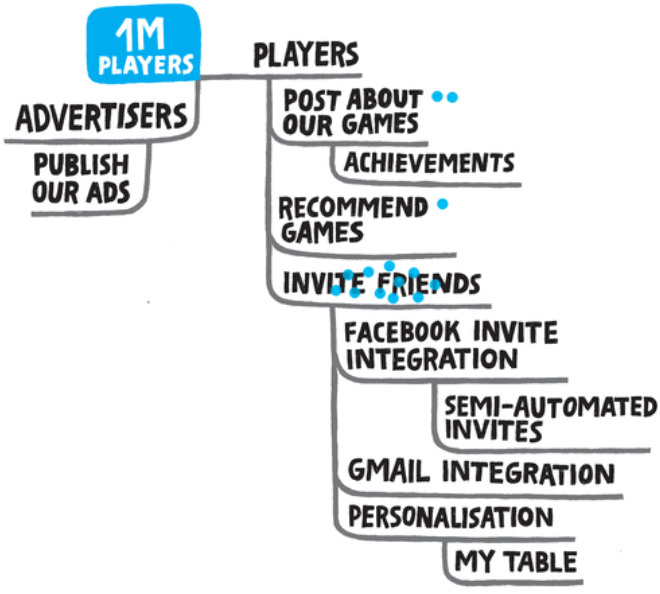
In uncertain cases, the only approach possible is to try something and see if it works. Senior people might have a good gut feel about something, and this should be enough justification to work on it. Some scope work will be experimental, and if it fails but helps us discover the right things to do it is still worth doing, though not at any cost.

Work with the group to define your learning budget: the maximum cost or length of tasks aimed at learning. On my projects this is often between one day and one week. Grow the deliverables so that each item is either justified by a validated assumption or fits into the learning budget. Ask the following questions to start a useful discussion on deliverables:

- What is the simplest way to support this activity?

What else could we do?

- If we're unsure about the assumption, what is the simplest way to test it?
- Could we test it without software?
- Could we start earning with a partly manual process?



If you can't define small experiments to test key assumptions, try user story mapping [Patton08b and Patton08c] or the hamburger method [Adzic12] to identify iterative delivery slices that could help you earn or learn sooner.

Ask: 'Are we sure that the assumption behind our #1 item is correct?'

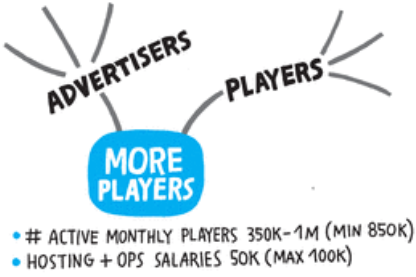
If the answer is 'No', find a way to test the assumption within your learning budget!

Fitting measurements into the map

In order to use an impact map as a delivery roadmap, we should have all the key information in a single place, so that we can use it to facilitate a good discussion while reviewing progress. This includes the business goal, key actors, expected impacts and deliverables, but also key metrics. There are four good ways of fitting metrics into the map.

Add bullet points

The first good approach is to list measurements as bullet points next to a particular map node or inside the node. The benefit of this approach is that it makes the distinction between measurements and actors or impacts quite clear, and keeps the map simple. The downside of this approach is that most mind-mapping software doesn't allow complex text formatting, so this approach might work well on a whiteboard, but not when you capture it electronically later. When doing bullet points, I often omit some information, such as where things will be measured. This could be included in the electronic version or an additional explanation.

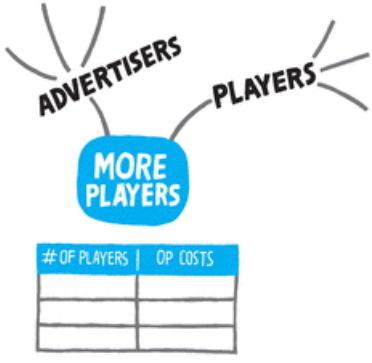


Rephrase map nodes

When there are only one or two key measurements for a node, we can rephrase the node name to reflect the key measurements. So instead of 'more players', we can say 'between 800K and 1M players over 6 months'. This works well for both the central goal and impacts. The benefits are similar to the bullet-point approach: the map is consistent and clear. The downside is that the map node text becomes larger, so this approach only works when there are a few key measurement points for each node in the map. When using this approach, I omit even more information than with bullet points to keep node text short.

Separate metrics table

If there are a lot of measurement points and you do want to show all the information about measurements (for example, if you're creating a document to be sent out to external stakeholders), it might be better to have a separate table with metrics included next to the map. The benefit of this approach is that a lot of information can be easily organised without over-complicating the map. The downside of the approach is that the map image on its own in no longer sufficient to facilitate future good discussions. When working with large whiteboards, I often use one half for key metrics and one half for the map – this keeps all the information visible to everyone.

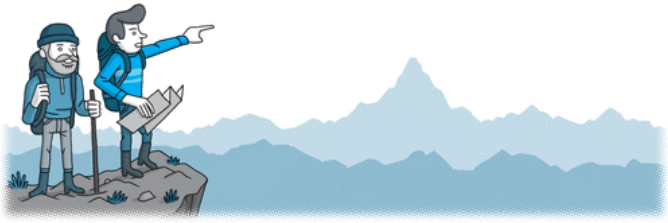


Attach additional map nodes

The approach suggested by Balic and Domingues in *Effect Managing IT* is to list measurements as additional nodes in the map. The benefit of this approach is that it is easy to include all the measurement information, and even organise measurements hierarchically. This approach works with any mind-mapping software, as metrics are just created as normal map branches. The downside of this approach is that the map can get a bit confusing if the structure becomes complex, as different concepts are mixed on same levels of the map. In particular, this starts becoming problematic for measurements assigned to impacts, not just to the business goal.


ADVERTISERS
PLAYERS
MORE PLAYERS
MEASUREMENTS
#ACTIVE MONTHLY
WHAT
GAME DB
WHERE
MORE PLAYERS
350K
CURRENT
800K
MIN.
1M
TARGET
OPERATIONAL COSTS
WHAT
HOSTING + OPS SALARIES
WHERE
FINANCIAL ACCTS
CURRENT
50K
MAXX
100K
TARGET
50K

Rinse and repeat



Impact maps and measurements provide great focus and alignment. Because of this, they carry the same potential pitfall as the number-driven failures in city building and agriculture that James C. Scott warns against. Scott shows how plans that narrow attention to a single outcome, typically one of high commercial interest, have "an extraordinary power to increase yields", but they have blind spots for anything outside the narrow field of vision, in particular long-term outcomes and third-party effects. Such problems typically stay hidden until they start to affect primary goals, at which point it is often already too late to fix them.

To avoid this pitfall and get the most out of impact mapping, use impact maps for iterative medium-term plans: product modules and project milestones. Between mapping sessions, watch out for unintended third-party impacts and longer-term effects that might have fallen outside the

narrow field of vision provided by a map. During delivery, remember that the connections in an impact maps are assumptions, which can be proven wrong. In very much the same way as you would use a real roadmap of an unknown territory, make a few moves and then check where you are. Measure progress and decide whether to keep going in the same direction or do something different. Once the first few items are delivered, measure the results. Deliverables that fail to produce results should point to invalid assumptions. Validated assumptions should justify further investment in the same part of the map. If the aim was to support players inviting their friends, see if they are actually doing this. If yes, you've de-risked the underlying assumptions so you can now safely invest more effort in this branch. Add more deliverables to the impact, and measure again. Remember to check if the higher-level assumption is correct. If players invite friends and they come to play our games, all is well. If invitations go out but new players aren't coming in as expected, then the whole idea of supporting invitations with further functionality should be questioned. Treat such things as failed experiments if you can. Seriously think about removing functions that fail to meet your expectations from the software.

Tom Gilb advises delivering projects in iterations that are not larger than 2% of the overall investment. Eric Ries advises scheduling regular pivot meetings to decide whether to stay on the same course or fundamentally change something. Agree up front with the group when you'll measure overall progress, for example every week or

every month, and try to identify the constraint and target intervals for those expectations.

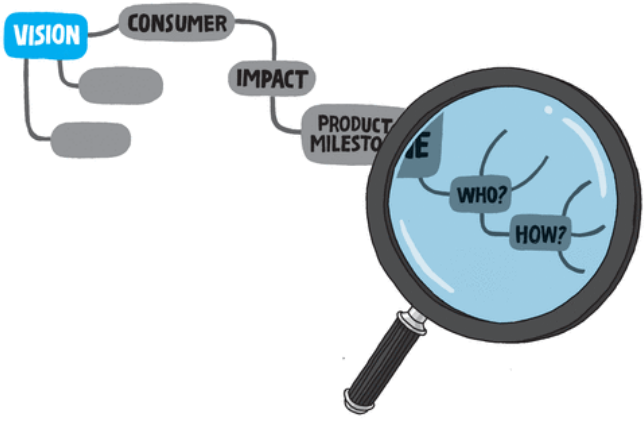
If the delivery fails to achieve the minimum targets, it's time for a strategy rethink. Discuss whether to change deliverables for an impact, or to work on another area of the map altogether. Consider whether there are long-term effects, off-stage actors or third-party impacts that are missing from your plans. If you fail to deliver several times consecutively, consider if the plan is realistic at all, and if you want to keep investing further or stop. Don't be afraid of revisiting key targets and measurements. Douglas Hubbard advises thinking about measurements iteratively, and adjusting what you measure based on initial findings.

Finally, if you've achieved the expected goal, stop. It doesn't matter if the entire budget was not spent – in fact it's even better. Once you reach your milestone, regardless of what was actually delivered in the software, you're done. It's time to move on to the next milestone, and do another mapping session with senior actors.

Large-scale maps

Larger organisations, or teams working on longer-term product development, might find it useful to create two levels of maps: one for the overall product vision and one for medium-term delivery. The overall vision map should capture desired longer-term effects and impacts on consumers. Deliverables on that map become product milestones, each with a separate lower-level impact map

when its time comes. Using two levels of maps can help track longer-term effects, and also split the work of a larger organisation into several parallel streams, each working on its own lower-level impact map.



Measure progress periodically against key milestone metrics.

If the delivery fails to achieve key targets, it's time for a strategy rethink!

Typical organisational mistakes

Here are a few things to watch out for when facilitating a preparation or mapping session.

Too many people

You can't get the wisdom of crowds on your own, but the more people you have in a workshop the more difficult it will be to facilitate it. This is particularly important for the preparation session, in which the aim is to nail down business goal and measurements. Everyone will want to have their say, and with 20 people in the room just doing the first round can take hours. Try to limit the first session to no more than five or six people, and ensure that the key business and technical people participate. If you have to work with a larger group, try splitting it into several groups that work in parallel and then present results. For the second session, you can have anything between ten and twenty as you can get the benefits of set-based design by diverging and merging. At that size, the workshop requires a dedicated facilitator. You don't have to hire a professional, but ensure that there is at least one person whose job it is to make the workshop run smoothly, not

necessarily to contribute to the map. The facilitator should keep track of time, prevent anyone from pushing their agenda too much and ensure that the right questions get asked at the right time.

Wrong people

The ideal group to define the goals and draw the initial roadmap is a mix of technical experts and business decision makers. Without decision-makers the exercise is pretty much pointless, because they have to define the key objectives and approve alternative solutions. Without technical experts the exercise might produce an interesting product or project definition, but we won't know if it is feasible or if

there are important technical risks to consider. Generally, you want senior people in the room, those whose opinion is respected and matters. I've done whole-team workshops where many people complained that it was too high-level for them, and that they just want a list of tasks to do. If people in the room aren't interested in what you're talking about, stop the workshop and find people who are higher in the organisation.

This is a problem often with mid-level management in larger organisations – these people are tasked with delivering a particular set of features and measured on that, so they won't see any benefit in coming up with alternative solutions, and won't have real power to make decisions.

Room layout

The ideal room layout is one that facilitates interaction and allows you to split people into groups to diverge and merge. The worst possible room layout is a theatre or classroom

with fixed chairs, because everyone will fall asleep. A boardroom where 90% of the space is taken up by a large table is pretty much useless. You're not negotiating a peace treaty, but trying to come up with a good project roadmap. Large tables encourage people to sit and play with their mobile phones while someone else is presenting, instead of participating. I've got into trouble with the facilities people several times for dismantling tables and pushing them to one side, but it was worth it. Give people room to move and collaborate and they will respond with more interaction and better ideas.

The room should also have plenty of surfaces for people to write on. If you don't have whiteboards, but there are large walls or windows, use whiteboard static film. It's a good idea to have a roll of whiteboard film with you in any case, to create more writing areas or capture discussion results.

Don't go for projector screens, or even worse large TV screens running software. If there are materials to present, print them before the workshop and hand them out to groups to analyse.

Typical facilitation mistakes

Criticising too early

To choose the right solution, you first have to have enough options. Ideation and divergent thinking, especially with several groups, bring great results quickly. I've facilitated workshops where we effectively saved 90% of the planned budget because someone came up with a completely different idea that would achieve the same impact. But in order to achieve similar results you have to facilitate a discussion about alternatives, even if most of them will never be implemented. Criticising too early can kill the discussion before people have come up with good alternatives. Even a

seemingly silly idea might inspire someone else to think of a good realistic alternative.

I often announce divergent and convergent thinking as separate phases, and tell the group that the goal of the divergent phase is to identify many possible ideas, without anyone criticising anything.

One big group

In a group there is typically someone leading. If that person is a strongly opinionated senior decision-maker, they will drive the discussion their way. Having everyone working in a single group is a good way to kill the creativity and feedback. Split people into multiple groups and you'll avoid the risk of a single person driving the agenda, and you will also get the benefits of set-based design.

Reverse-engineering an entire shopping list

I've facilitated quite a few workshops where the team already knew what they were supposed to do, or at least thought that they did. We started with huge shopping lists of features, sometimes hundreds, and in two days of impact mapping agreed on much better solutions. Throwing away the whole shopping list at the start can cause a lot of objections, as some people in the room have probably spent a lot of time compiling it and there are probably some good ideas in there anyway. I often use a shopping list to start a discussion, so that we're more effective than starting from scratch. I use some key items on the shopping list to draw the map skeleton, and then push people to think about alternatives.

Then again, reverse-engineering the entire list is a horrible waste of time, especially with senior stakeholders in the room. As a facilitator, try to spot when there is

enough information on the starting map to inspire people to look for alternatives. A good trick is to split the group into several smaller groups and let them choose parts of the shopping list to reverse-engineer into a map skeleton. Merge them in fifteen to twenty minutes and compare results. The discussions the individual groups have had should prime them for the next step, and a merged map combining several options from different groups should be enough to inspire a discussion on alternative actors and impacts.

Going into too much low-level detail early on

With iterative delivery and a roadmap, we only need to have the big picture in sight and know the next few steps. Expanding the map too much, in particular generating scope ideas for all level 3 items, is overkill. Making people

go through the list and prioritise the ideas is a complete waste of time. If you have senior decision-makers in the room, they should be using their time for something more valuable.

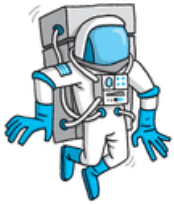
I generally ask people to focus on actors and impacts first, making sure that they are not trying to sneak in scope as business capabilities. This will give you fewer items by an order of magnitude to consider when prioritising and comparing than if you built the entire scope.

Typical mapping mistakes

The Jumper

One of the most common mistakes is to jump over levels in the map, often skipping over the impacts and linking technical scope directly to actors. This is problematic because it does not provide a good framework for divergent thinking nor does it bring out assumptions well. If you spot teams jumping over levels, get them to explicitly list the missing information.

The Astronaut



Another common mistake is to map out goals and impacts without good metrics – like a weightless astronaut in space, we won't be able to get good feedback on movement. Having measurable objectives is essential for deciding whether something is realistic and if we're making good progress or need to rethink our strategy.

The Surrealist

Objectives with unrealistic or non-actionable metrics will not facilitate a good mapping session. Once you've identified the key measurements and defined your milestone,

make sure that the people in the room can actually deliver those results. If not, either redefine the objective or replace the people in the room.

The Shopper

Going into too much detail early on, unless you need to fix scope up front (which defeats the point of iterative delivery), is just a waste of time. You are much better off spending that time on identifying all the actors and impacts and prioritising them. When groups start with a shopping list, sometimes they insist on mapping out all existing ideas. Stop them when the key ideas are mapped out, since this will be enough to facilitate a good discussion.

The Optimist

I've never been involved in a project that had no potential obstacles. Still, plenty of teams I worked with focused only on satisfying their users' needs or supporting positive activities. If the team is working on a map that does not have any actors or impacts that could prevent delivery, ask them to think about this – they will then often identify constraints on key deliverables.

The Dreamer

Impact maps make us think about actors and impacts on their behaviour, and sometimes groups take this too

far. They starting to plan for solving all the needs of an important user, regardless of whether this fits into the current milestone or not. Avoid spending too much time discussing impacts that do not contribute to the overall goal. Either find a way to justify them or discard them.

The robot

The goal of a mapping session is to define a delivery roadmap. If you're working in software delivery, most items in the scope will be software implementation stories and tasks. However, it is highly unlikely that implementing software is always the right solution – especially for learning activities. If all the items on a map are technical, ask the group to identify how they would test assumptions without implementing software or impact actors' behaviour in a different way. This often creates a breakthrough in the discussion.

The octopus

It's not critical for a map to have a single measurement: it is perfectly valid to ask for an increase in sales and a reduction in IT cost. But trying to map too many things at once will kill the focus of a mapping workshop. Instead of trying to achieve several things at once, try splitting them into several milestones.

Where next?


WAIT... WHAT NEXT?
WELL, THIS IS MY FLOOR, BUT YOUR JOURNEY CONTINUES.

www.impactmapping.org/engage

www.impactmapping.org/poster

www.impactmapping.org

http://gojko.net/impact

Bibliography

Essential introduction to background techniques

- [Balic07] Effect Managing IT by Mijo Balic and Ingrid Ottersten, Copenhagen Business School 2007, ISBN 978-8763001762
- [Brinkerhoff94] The Learning Alliance: Systems Thinking in Human Resource Development, Robert O. Brinkerhoff, Jossey-Bass 1997, ISBN 1555427111
- [Gilb05] Competitive Engineering: A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage, Tom Gilb, Butterworth-Heinemann 2005, ISBN 978-0750665070
- [Matts11] Feature Injection: three steps to success by Chris Matts and Gojko Adzic

References for the numbers from the introduction

- Troubled £12bn NHS IT system to be scaled back, BBC News, 6 December 2009

- NHS pulls the plug on its £11bn IT system, The Independent, 3 August 2011
- Costs of US piloted programs by Claude Lafleur, 2010
- Also see [McManus08]

Other books and online articles

- [Adzic12] Splitting user stories: the hamburger method, Gojko Adzic
- [Berkun05] Art of Project Management, Scott Berkun, O'Reilly 2005, ISBN 978-0596007867
- [Brown09] Change by Design: How Design Thinking Can Transform Organizations and Inspire Innovation, Tim Brown, Collins Business 2009, ISBN 978-0061766084
- [Cohn04] User Stories Applied for Agile Software Development, Mike Cohn, Addison-Wesley Professional 2004, ISBN 978-0321205681
- [Cohn06] I didn't know I needed that, Mike Cohn, Better Software Magazine February 2006
- [Gray10] Gamestorming: A Playbook for Innovators, Rulebreakers, and Changemakers, Dave Gray, Sunni Brown, James Macanufo, O'Reilly Media 2010, ISBN 978-0596804176
- [Harford11] Adapt: Why Success Always Starts with Failure, Tim Harford, Farrar, Straus and Giroux 2011, ISBN 978-0374100964

- [Hohmann06] Innovation Games: Creating Breakthrough Products Through Collaborative Play, Luke Hohmann, Addison-Wesley Professional 2006, ISBN 978-0321437297
- [Hubbard10] How to Measure Anything: Finding the Value of Intangibles in Business Douglas W. Hubbard, Wiley 2010, ISBN 978-0-470-62568-2
- [Klein99] Sources of Power: How People Make Decisions, Gary Klein, MIT Press 1999, ISBN 978-0262611466
- [Lapouchnian05] Goal-Oriented Requirements Engineering: An Overview of the Current Research, Alexei Lapouchnian, Department of Computer Science, University of Toronto
- [Liedtka11] Designing for Growth: A Design Thinking Toolkit for Managers, Jeanne Liedtka, Tim Ogilvie, Columbia University Press 2011, ISBN 978-0231158381
- [McManus08] A study in project failure by John McManus and Trevor Wood-Harper, BCS, The Chartered Institute for IT, June 2008
- [Patton08a] I don't know what I want but I know how to get it, Jeff Patton, 2008
- [Patton08b] User Story Mapping, Jeff Patton 2008
- [Patton08c] The new user story backlog is a map, Jeff Patton 2008
- [Pixton09] Stand Back and Deliver: Accelerating Business Agility by Pollyanna Pixton, Niel Nickolaisen, Todd Little, Kent McDonald, Addison-Wesley Professional 2009, ISBN 978-0321572882

- [Ries11] The Lean Startup, Eric Ries, Crown Business 2011, ISBN 978-0670921607
- [Scott99] Seeing Like a State: How Certain Schemes to Improve the Human Condition Have Failed, James C. Scott, Yale University Press 1999, ISBN 978-0300078152
- [Shore05] Beyond Story Cards, James Shore, 2005
- [Sibbet10] Visual Meetings: How Graphics, Sticky Notes And Idea Mapping Can Transform Group Productivity David Sibbet, John Wiley and Sons 2010, ISBN 978-0470601785
- [West11] Water-Scrum-Fall is the reality of Agile for most organisations today, Dave West
- [Yu11] Social Modeling for Requirements Engineering, Eric Yu, Paolo Giorgini, Neil Maiden, John Mylopoulos; MIT Press 2011, ISBN 978-0262240550

About the author

Gojko Adzic is a strategic software delivery consultant who works with ambitious teams to improve the quality of their software products and processes. Gojko won the 2012 Jolt Award for the best book, was voted by peers as the most influential agile testing professional in 2011, and his blog won the UK Agile Award for the best online publication in 2010. To get in touch, write to gojko@neuri.co.uk or visit http://gojko.net

Legal stuff

Impact Mapping:
ISBN: 978-0-9556836-4-0

Author: Gojko Adzic
Copy-editor: Marjory Bisset
Design and layout: Nikola Korac
Published: 01 October 2012
Epub Version: 1.1

Provoking Thoughts
Menzies Suite A,
1st Floor 62 Goldsworth Road
Woking, Surrey
GU21 6LQ
United Kingdom