more onion

Management

PROCESS

that delivers real world

IMPACT

What we
DO

At more onion we create effective strategies, convincing creatives and cutting-edge technology. Our focus is on e-campaigning and digital fundraising. We always strive to deliver maximum impact with the resources available, regardless of the tactics chosen.

What we've learnt from many years of agency work and successfully delivering hundreds of projects is that maximising impact for our clients is very much related to having a good process in place. Good process means managing workflow, resources and constraints. But above all it means managing the complexity of knowledge work.

In this short document we would like to take the opportunity to highlight how we think our management process can deliver results for your organisation.

Our headline
RECOMMENDATIONS

1. Don't set out your requirements in stone at the beginning of the project - encourage new ideas throughout and your objectives will be met faster, more efficiently and at less cost.

2. Don't use an RFP process - instead hire an agency that has delivered projects in an iterative manner (or design a pilot project to test out your chosen agency first). Otherwise you'll only find out at the very end of the process if your chosen agency will go over budget or fail to meet your requirements.

3. Pick an agile process to deliver your projects - that way, if a designer has a great idea for a solution part way through the project it can still be incorporated. But if the specs have been agreed at the outset, you won't benefit from that innovation and improvement.

4. Ensure that working parts of the whole project are delivered as often as possible through the project so you can get "real world" feedback on your project while it's still being developed. For example, launch a version of your website really early in the process while it's still a little rough around the edges to see if your assumptions about your audiences are correct. Then optimise the site based on these learnings.

5. Fail cheaply and fail early in the process - design tests to see whether the assumptions made at the start of the project were right. This way, if something isn't working, you can find out at the earliest possible point, rather than at the very end of the project.

6. Next time you hire an agency and you're thinking about using an RFP process, ask yourself this question: will I really get all the evidence I need to decide who to hire from this process, or will I just get a slick sales presentation?

We explain why in a lot more detail below.

The nature of complex **PROJECTS**

Creative work, digital communications and software development can be summarised as "knowledge work". Knowledge work usually requires people to solve problems and figure out what to do to make a project successful as part of their work.

And one thing is certain: ***knowledge work is complex.***

In the context of campaigning, creatives and designers have to come up with ideas and solutions that will convince and persuade people to take an action.

Digital communications often means learning how to do your job multiple times over the course of a year or two as new channels emerge, new tactics are developed and existing channels stop working.

And software development involves figuring out how to best serve users of software and websites, and creating and managing hugely complex systems and projects.

This complexity leads to a lot of challenges and most of these challenges are related to how these projects are managed.

Knowledge work such as the examples outlined above can be hard to predict, it has significant risks inherent to it (what if the brilliant creative idea actually doesn't get supporters taking action?) and can't be easily replicated (just because a tactic worked for one organisation, doesn't mean it'll work for a different one - campaigns asks vary from organisation to organisation, as does the profile of their supporters).

And because knowledge work is always operating in a changing environment, the scope delivered can never be 100% flawless. With a website, for example, what's good practice at the outset of the project may have changed just 3 months later because your fundraising database has changed, or a new channel has emerged (the mobile market grew from nothing to being of crucial importance in just 1-2 years for example). Due to the inherent complexity of the work, there is no guarantee of success in each and every single small aspect of the project.

So knowledge work has to be able to deal with changing circumstances, the very circumstances that make it unpredictable.

Traditional project
MANAGEMENT

Traditional project management suggests following a simple recipe to ensure your project is a success:

1. Initiation
2. Definition & Planning
3. Launch & Execution
4. Monitoring & Control
5. Closing & Evaluation

This process is fine if absolutely everything goes according to plan. But if circumstances change during the stage when your new website is being built - for example new insights emerge from showing real users the half-finished website which mean a change of plan is needed - then under this form of project management you will only find out towards the end of the process. And this will lead to substantial delays and potentially significant cost overruns for the entire project.

Here are some examples of things that might change during that stage:

• A key stakeholder in the project remembers there's something really important that must be included in the website;
• You win a campaign which means that some fundamental changes are required to the project scope;
• Updates to key software mean changes are required to the design to ensure functionality on all devices and browsers.

How many times have you heard about projects that have gone substantially over budget? Or failed to meet their objectives?

That's because, by following a rigid process such as the one above, without responding to changing external circumstances, projects either failed to respond to those changed circumstances and were effectively obsolete at the time of launch, or they were very late because the rigid nature of the process prevented a reprioritisation to allow for the change in circumstances.

Moreover, by not setting requirements in stone at the beginning of the project and instead encouraging new ideas throughout, the objectives can often be met faster, more economically and with greater impact. Hiring experts in the field to help work out what those requirements are will potentially allow you to find smarter solutions than someone working at the organisation - it's good to have an external pair of eyes working with you who is close to your target audience.

The good news is that there are many management methodologies that focus on managing complexity, unpredictability and changing circumstances.

Later in this document we will highlight some techniques that we use at more onion.

The essence of these new techniques of management boil down to one thing: managing risk.

The risks inherent in
PROJECTS

A complex project is always risky.

Generally projects that fail are seen as failures because they have not met your objectives or because they've ended up being more expensive than expected. Both of these create risks for the entire organisation. Good management practices can't prevent risk entirely, but they can minimise it as much as possible.

If the team working on a project is allowed to do the following things, then these risks will be minimised:

- gain a deep understanding of the objectives and explore different ways to achieve them (e.g craft a strategy before deciding on the tactics)

- respond to changing requirements and circumstances quickly

- requirements are not set in stone at the beginning of the project - instead, changes throughout the project should be encouraged if the new ideas will help achieve the objectives faster / cheaper / better

- the objectives are used to create a hierarchy of priorities for the project which determine which features are developed in which order. These priorities can change every few weeks, once some features are complete or if priorities need to change due to a change in circumstances

- fail cheaply and fail early in the process; design tests to see whether assumptions made at the start of the project were right - if something didn't work, learn from it. If this approach isn't taken, and the hypotheses at the outset are assumed to be correct, the earliest point at which it's possible to discover if any of those hypotheses were incorrect is at the end of the project.

- deliver working parts of the whole project in an iterative manner as often as possible during the project. This way you can get "real world" feedback on your project during the development phase. If you "launch" often you can distribute the risk and you can optimise based on the learnings from previous launches.

- put people before process and documentation. In the area of knowledge work you have to accept that everything comes down to people and their ability to perform together. Process should help and support but never make people feel dull or repressed. People are more important than documentation.

- deliver best results through self-organisation. Allow the different teams to identify possible solutions to the problem, experiment with these solutions (and fail cheaply) and empower them to self-organise. Only then can the team deliver innovation and knowledge work that really meets the objectives.

- continuously improve themselves and the delivered results. The ability to learn and improve within a project is key, since the discovery of new opportunities, threats and changing requirements will only happen during the project.

These principles are usually applied if agile management is practiced.

Why is a Request for Proposal (RFP)
NOT SUITABLE?

The most common way of hiring an agency to deliver a piece of work is the "RFP process". The organisation puts together a document with some rough requirements, sends it to a number of agencies and the agencies send back a proposal.

There are many reasons why this process usually delivers poor results.

An RFP process strives to compare costs. The problem is that the range of solutions presented to meet the scope outlined in the RFP will differ vastly. As a result, a direct cost comparison is not really possible. And it's only towards the end of the project that you'll find out if the agency will go over budget or not.

Since the scope is described and "agreed upon" up-front there is no point in the process to discover requirements where the entire team on the agency side can work with the client to explore potential solutions. However, this process is crucial for every complex project.

The upfront specification of requirements often means the team will work in an incremental fashion and deliver one piece of the puzzle at a time. This approach increases the risk of the project because the big picture can't be tested and validated until the very end of the project. For example, you might want to test whether the most important user journeys for the audiences which were determined at the outset of the project are working properly - it'd be good to find that out as early as possible in the process, not right at the end, so that changes can be made without incurring additional costs.

So as a result RFPs mean that agencies submit pre-defined solutions or ideas that have not had a reality check through an iterative process.

At times an RFP seeks to explore different options and even entirely different ways to solve the problem. However, the best solutions can't possibly emerge before the project starts. As we've already explored, the circumstances of the project will change, requirements will change and new opportunities may come up.

An RFP process is often adopted to make the hiring process more objective and easier to justify in front of other stakeholders and senior management. However, choosing an agency based on the proposal they submit is like buying a house based on the sales brochure. It merely judges the sales abilities of the agency in question. You shouldn't pick an agency based on gut feeling either. Or on their willingness to spend a lot of time working for free at the outset to respond to the RFP, and then cut corners during the delivery of the project once the contract has been won in order to compensate for the time that's already been spent winning the project.

To sum up:

1. RFPs fail to compare cost properly because they are not comparing like with like
2. RFPs force agencies to come up with solutions without going through an iterative discovery process
3. RFPs fail to deliver different potential solutions to the problem because the best solutions will only emerge during the course of the project
4. RFPs often lead to incremental project delivery which leads to high risk because the big picture is only tested (and seen) at the end of the project
5. The upfront description of the scope leads to less flexibility during the project to react to new challenges, opportunities or the ever-changing circumstances

Instead of RFPs we advise you to have a conversation with those agencies you are seriously interested in. Outline the constraints, the objectives and ask them to outline the process they would apply to meet your objectives.

You can either hire an agency that has delivered similar projects in an iterative manner before (and has delivered the objectives), or you can create a test pilot project.

Come up with a pilot project that helps you learn and discover valuable lessons for the main project. Sometimes it can also be the first 1-2 iterations of the big project. Then simply hire the agency you think will deliver and see how much they can help you reach your objectives.

If your budget allows it you can also hire two agencies to deliver the same project or other small pilot projects - all of them designed to explore what may be possible solutions to your problems.

That way, you will actually really see if the agencies you are thinking of hiring:

• deliver objectives
• stick to the budget
• meet deadlines
• and just as importantly, what they're like to work with.

The basics of
AGILE

We think agile project management is the best way to manage complex projects.

Agile management has it's origins in software development but quickly spread to other industries through the startup communities of Silicon Valley. The problem was that the traditional project management methodologies were failing. Huge software projects were delivered on time and delivered on spec but failed to meet the users' needs.

In a fast paced world like ours you can't put together a team, hand them a 50 page specification, let them work in a basement for a year and then expect the result of this project to deliver the desired impact.

"Things change fast in the world of the internet."

This is why agile was born, to conquer complexity and ever-changing requirements.

While the general idea of agile management is only defined rather vaguely there are several methodologies such as Scrum, Xtreme Programming and Crystal Methods. They all use a slightly different approach but prioritise iterative project delivery and exploration over fixed documentation and "big launches".

They all use "cycles" of iterative delivery that can be a few days long or a few weeks long. At the end of every cycle the team delivers a fully functional version of a specific solution to a user need. For an example instead of developing the entire organisation's website and then launching it, the website could be launched very early in the process while it's maybe a little rough around the edges.

But then it gets better and more polished at the end of every cycle.

The advantage of this approach is that the organisation as well as the agency can learn from how people are interacting with the site, what stakeholders say (feedback) and make user testing more "real".

At the beginning of every new cycle the organisation and the agency can decide what to deliver as part of the next cycle, and if they want to they can even change their priorities entirely. This way the agency can incorporate these changing requirements

along the way while still remaining within budget.

Essentially the risk of failure is reduced to failing at individual iterations. If something doesn't lead to measurable progress (according to the Key Performance Indicators) it can be changed in the next cycle. However, at the end of the entire process you will certainly have a result that has had a good reality check.

On top of that agile delivers some level of predictability because the scope will only be defined and estimated for one cycle - which is a much smaller unit. Through experience and over time every team will develop a certain rhythm that can be used to predict the amount of work that is likely to be delivered in one cycle.

This method allows for better planning and since the team guarantees to have a functional version at the end of each cycle the worst case is that some aspects are not delivered in this, but rather the next cycle.

Planning as part of agile management always happens from the user point of view. So called "user stories" are formulated and describe exactly what the user can do or will experience at the end of the cycle.

This way the team is fairly free to explore different options on how to meet this user need. It also means that there is a strong focus on delivering something that really solves a user problem.

In short: the key to delivering complex projects is to apply an "agile" process. Agile management provides a framework that empowers the team to deliver the (almost) impossible.

At more onion we use agile cycles and a mix of some of the methodologies (SCRUM and Xtreme Programming) for the product team as well as the project team. The product team uses a longer sprint cycle to reflect the release cycle of products while the project team uses a weekly cycle to turn around projects as quickly as possible.

The basics of
KANBAN

Methods like Kanban and "Lean Development" are also part of the "agile" family but don't necessarily use iterative cycles of delivery (or "sprints" in agile terms). Kanban has its origin in manufacturing and is a method to streamline workflows and reduce the problem of bottlenecks in a workflow. The method inspires team collaboration and problem solving through transparency and visualisation of the work (usually through a board on the wall).

Kanban manages to streamline a team in such a way that the time of delivery of every element is highly predictable. By imposing a limit on the work that can be "in progress" at once it encourages the team to collaborate to finish individual tasks instead of working on many things simultaneously. This aspect of kanban increases the productivity of a project team and the quality of the results.

Kanban also limits the work of tasks that are planned and scoped out. The result is that only those tasks will be described in detail that the team is just about to work on. The priority and new tasks that will fill up the "backlog" (all user stories that are ready to be worked on) are decided in a meeting with all key stakeholders once every week or so, depending on the organisation.

Since Kanban does not work in cycles as such it can be used for work that has to be processed as it comes in. At more onion we use Kanban to manage support related work and the project team also switches to "kanban mode" during the launch stages of a project.

During the launch by using Kanban we frequently handle issues a few minutes or hours after they have been reported.

UX, *creative and design*
PROCESS

When working on creative components of projects or User Experience Design a strict iterative process is not always possible. Creative process requires a fairly open period of research, interviews, observation, gathering of use cases and prototyping.

At the end of this open process the ideas become more specific and the good ones are taken over into the iterative development.

At the end of every iterative cycle we measure if our ideas have had the desired impact or not.

We validate experience design assumptions through user testing, measuring the key metrics and at times even role playing. For optimal results user experience experts and the creative team have to be involved in the entire process, not just at the very beginning.

For visual design we apply the same process. We start off by creating mood boards, testing a few different directions and then choose drafts we finalise. During the development process designers frequently check in and actively participate, tweak and optimise.

To us creative concepts "work" if we can measure their impact. An online action is considered "good" if it convinces people to take action and meaningfully engages them. A donation form is beautiful if a lot of people make donations using it.

Our creative process is partly inspired by techniques formalised in "Design Thinking", a methodology that puts user centered design at the heart of everything.

Design thinking suggests a process should adopt the following ideas:

- Empathise, learn about your audience and their needs
- Define a point of view that you will use in the design process that clearly shows user needs and problems
- Ideation, come up with many potential solutions to the problem, explore
- Prototype to learn more about potential solutions
- Test the prototypes with your user group and target audience

At more onion we usually cut this process short and allow it to fade out into the iterative agile cycle. It starts off with research and understanding the user, then we prototype potential solutions that are tested with those users. As soon as the prototypes are refines we start working on the the first possible solution - the "Minimal Viable Product".

The idea of the
MINIMAL VIABLE PRODUCT

The idea of the Minimal Viable Product (MVP) has it's origins in the "Lean" methodology. The basic principle is to work towards the first possible output, product or solution that is good enough to get started with real users - for example a first version of a website or landing page.

Often this version does not have a lot of complicated functionality but instead focuses on making the first 1-2 most important use cases possible. For an example an online petition could be launched as soon as there is some basic content on there, a form (that saves the captured data), tracking and a thank you page once people have completed the form with some share buttons.

The second version of the page would have a more refined visual design, a progress bar and live ticker to show how many people are taking action right now, there would be more advanced reports based on the tracking data and split testing of the images used on the page to see which gets the best conversion rate.

The MVP is used to focus on a first usable version to begin with, to kick off an iterative process with real users and real data.

What's the
CONCLUSION

At more onion we are dedicated to delivering the maximum impact with the resources available. The process and methodologies described in this document allow us to deliver just that.

If you have any questions feel free to send Florian an email or give him a call!

Florian Engel
florian@more-onion.com
+43 69917 12 5844
Skype: flo.engel
Hangout: moreonion@gmail.com