Required reading: Getting to grips with NFRs

In this article, Mike O’Hara, publisher of The Trading Mesh talks to Ash Gawthorp of The Test People, Matt Barrett and Olivier Deheurles of Adaptive Consulting to find out just how critical it can be for companies to take a prudent and careful approach when setting non-functional requirements for new technology.

Introduction
Non-functional requirements can make the difference between game-changing technology and expensive flops. A non-functional requirement doesn’t determine if the product will work, but it can determine if the product will work well. In one widely cited study, German and Japanese engineering researchers noted the failure of the Mercedes Benz A-Class to pass the ‘Moose Test’ as an example of what can happen when NFRs are not adequately considered. The Moose Test, which has been around for decades, involves a vehicle making quick turns to avoid an obstacle, as if a car suddenly had to avoid a moose. German auto engineering has a reputation for excellence, but in this instance, which took place in the late 1990s, the Mercedes overturned when put to the test. So how can businesses make sure their technology will pass whatever equivalent Moose Tests they’ll face? The answer appears to have as much to do with mind-set as with technological expertise.

Holding the line
During those critical moments when requirements are being defined, Ash Gawthorp knows that businesses need to have one eye on the now and one on the horizon. Setting and defining non-functional requirements at the start of a project often doesn’t give you immediate value but pays dividends a little further down the line as the software matures.

“Setting non-functional requirements is all about considering the characteristics that don’t immediately spring to mind from the product point of view, but must be defined, and embedded in the approach from the start to avoid it biting you further down the line” said Gawthorp, technical director at consulting firm The Test People.

In the case of functional requirements, it’s all about the present. As Gawthorp said, something either works or it doesn’t, and most people will be able to spot the difference. But with NFRs, it’s not always clear how important a requirement will become later on. “It’s the characteristics and features that are either invisible to you as a user, or only impact the user experience under specific scenarios - hardware failure or high concurrent user load for example” he said. “And if you don’t invest time and effort into it, then the risk of something going dramatically wrong increases significantly over time.”

All businesses are under pressure to get new features and functionality out of the door in their desire to get ahead of the competition. This, Gawthorp said, is often at the cost of good engineering practices resulting in technical debt “if you are under pressure to get new features out of the door faster and faster, then often a few corners have to be cut… if you don’t invest the time and effort to tidy up and refactor you end up building this ever increasing debt of technology that has to be addressed at some point.”

His firm sees this behaviour frequently. It requires strong leadership in IT to push back on the business and make them aware of risk of technical debt building up, and to ring fence budget to ‘shore up’ any corners cut to meet the feature requirements. Failure to do so risks putting the entire platform at risk of catastrophic failure. “It’s essential for IT to ‘hold the line’, they have a responsibility to ensure the platform continues to adhere to NFRs, and where it starts failing time must be spent to remedy the problem early, and not ignore it folding under the pressure of every increasing feature requests.”

1 The study cited can be found here: http://www.i3s.unice.fr/~mirbel/enseign/article-1.pdf
Many people believe that by thinking about performance and scalability, they’re doing enough to future-proof their efforts. Performance and scalability definitely matter, but there may be a dozen or more non-functional requirements that can be equally important. Response time and throughput were a couple of examples he gave.

“If you are under pressure to get new features out of the door faster and faster, then often a few corners have to be cut… if you don’t invest the time and effort to tidy up and refactor you end up building this every increasing debt of technology that has to be addressed at some point.”

Ash Gawthorp, The Test People

Setting good NFRs also means having the ability to test and monitor them. This is where technologists can help a business make better decisions. Say a business owner wants to add a particular feature to a product or system. With the right NFRs in place - and the means of measuring them - product developers are in a position to explain what impact the new feature will have on performance.

Sometimes, the impact of a single change is negligible but a series of small tweaks can collectively take a toll over time. “Often what causes failure in systems are not one huge single change or event when people can say, ‘Okay, we’ve done this and it’s suddenly fallen off the edge of a cliff’,“ Gawthorp said. This can be especially true when companies launch a system that fails to perform in the way intended or do everything the architecture was designed for. “As times goes on more features and functionality are added, without investing the time to address tech debt the purity of the design degrades, resulting in poorer performance and elongated times to add new features with greater risk,” he added.

Another key factor: money. Too often commercial executives view technology as a matter of cost rather than revenue. “At the other end of the scale is where an organisation sees that it’s not just setting NFRs, but ensuring that a system adheres to them as a real profit centre,” Gawthorp said. If business owners consider the amount of money they make as proportional to the number of users they have on a system, adding users and delivering a good service with the same cost or infrastructure becomes a major windfall.

The cost-revenue prism is just one aspect where a company’s mind-set is important. In the context of NFRs, the wider nature of the business-IT relationship becomes paramount. How business owners communicate and work with developers can make all the difference.

The negotiation

“There is a gap, as there always is, in communication between business and IT,” said Matt Barrett, a director at technology consultative group Adaptive. “A lot of the time when we get requirements, we get them without context.”

For instance, a company may say a system needs to process an event in a certain amount of time or handle a specific load. But it will neglect to give the business context, so IT will not have any understanding of how that requirement may change over time. Barrett’s advice: spend time on understanding context.

“If we just all shared the same context about the use of the application and the business case for it, the non-functional requirements would be obvious.”

He suggested the reluctance of IT technologists to often engage with the business properly may lead to an unhelpful insistence that they are given concrete numbers for NFRs. “It would be much better to just understand the business case and then it would be apparent what the non-functional requirement was,” Barrett said.

Gawthorp of The Test People describes the early stages of a project in terms of a negotiation.
“The negotiation implies two teams set up across the table from each other, seeing who will give up the least,” he said. But it doesn’t have to be adversarial. It should just be an open discussion where both sides learn a great deal. “When it works well, the business is able to explain to technology why it is that they feel this pain and why it is that something needs to be a certain way. Conversely IT is able to explain to the business why something is so much harder or riskier to do it in one instance when it may be trivial to do it in another way,” he said.

To make that negotiation successful requires communication and that in turn depends on at least two things: a common language and trust.

“There is a gap, as there always is, in communication between business & IT. A lot of the time when we get requirements, we get them without context.” Matt Barrett, Adaptive Consulting

Negotiations sometimes don’t get off the ground because IT developers are not aware of NFRs in the first place. Suddenly, a business owner wants to know why a function doesn’t take place quickly enough and frustrated developers respond that no one ever specified that time requirement.

“Part of the challenge is that common language,” Gawthorp said.

The ability to speak the same language can depend on who is at the table. For instance, he said development and test teams often are more involved with the business because they spend more time understanding business requirements and turning them into software; further downstream, teams such as deployment, infrastructure or networking tend to have less involvement with the business, although the areas they manage can affect important product or service features such as resiliency, robustness and security.

“The conversation needs to happen where IT can explain to the business what they can do and can’t do, what’s easy and what’s hard for them, and to see how everybody can move forward,” Gawthorp said.

The language gap may be so pronounced that sometimes, one side of the table is not clear on whether the other side wants something in the first place. So many applications don’t fulfil NFRs because developers actually were never told of NFRs by the business. And for product managers, it is not always clear whether they should be initiating the discussion about NFRs.

A matter of trust

Beyond the common language issue, there is a more fundamental factor: trust. Both sides need to be transparent about how important different requirements are. That’s where face-to-face discussion becomes paramount. If a business owner starts out saying that a certain requirement is ‘non-negotiable’ only to become more flexible when the development team says it can’t be done, it reflects a lack of honesty in the conversation.

Sometimes requirements are just handed down from on high. In such cases, if they are onerous, IT staff may not take them seriously or they may simply say the requirements are just not possible.

The main word that Gawthorp stresses is collaboration. He imagines a diagram with business owners on the left hand side and various technologists, each of whom has a part to play in the product pipeline, on the right. “There are often small changes that each of those teams can make, that will make the lives of the upstream and downstream teams far, far easier, but frequently don’t happen,” he said.

Sometimes that happens because teams work in silos. Other times, the common language factor is at play. “Really a conversation needs to go on around what is realistic and what is achievable,” he said.

Genuine collaboration between IT and business is actually a relatively recent phenomenon. “It’s only recently in the last 10 years or so where IT and technology has been able to open the box a little bit and let the business have a look inside,” Gawthorp said.
Olivier Deheurles, a co-founder and director at Adaptive, said it can be important to have at least one technical person working within the business team and helping to shape NFRs.

When a dialogue is working well, a business owner will be able to be more transparent when discussing the importance about certain requirements, or even the lack of immediate clarity about how precise they need to be. At the same time, IT staff will feel they can be honest about what they can and can’t deliver. That fosters the conditions for practical solutions such as having a fast feedback loop.

“It comes down to being agile,” Deheurles said. “It is not about building massive specifications up front and knowing everything.”

**The irresistible force paradox**

To illustrate what collaboration doesn’t look like, Gawthorp described a scenario where IT asks a business owner for an NFR on a log-in time. “IT says, for instance, ‘What is an acceptable time for a user to log into this system?’ the business’ response is, ‘As fast as possible’.

Alternatively, if a business owner suggests a time that would be prohibitively expensive to achieve and is therefore unrealistic, it can kill a conversation stone dead. In those scenarios, IT often will just go away and do whatever it thinks best knowing it can’t achieve it.

“Frequently an NFR will be provided with little or no investigation into the business benefit realised, say ‘ninety percent of logins must take under five seconds’, five seconds may require a huge engineering effort, but eight seconds is easily achievable - questioning the business value is important to avoid wasted effort,” Gawthorp said.

Despite the precision required to trade successfully in today’s markets, pinning down hard numbers can actually be problematic. In low-latency systems for example, it can be very difficult to specify what is the appropriate NFR to be targeting in terms of what is realistic and what is gold-plating.

In the basic example of the log-in time that Gawthorp gave, it may come down to whether to invest in lowering the overall time, or in having a slightly longer login but eliminating outliers where users are kept waiting for long periods of time.

“It’s almost a bartering system essentially, a give and take between what the business is looking for and what is realistically achievable and understanding those points.” He adds: “It’s really a question of understanding where the pain is and where to invest.”

This can involve weighting certain NFRs differently. But even if companies can have those honest, transparent conversations about trade-offs, it is still not a guarantee of success. They also need to have them early on in the process. Architectural decisions at the beginning of a project can have huge implications later in terms of scalability or any number of other NFR factors.

After all, it is rarely the case that there is one clear-cut way to meet a goal. Barrett said: “There are many different ways to achieve something when you are building these platforms but they all have trade-offs and the non-functional requirements are what guide you to make the right trade-offs.”

**It’s not all about the numbers**

Sometimes, getting the NFRs right can be more about feeling than numbers.

“It doesn’t always require specifying an exact number,” Barrett of Adaptive said.

A company need not create an iron-clad service level agreement and spend time and energy forcing a provider to adhere to it strictly. A more sensible approach could involve just building
the metrics and looking at the trends over time. “Don’t say the number. Say that you want to be able to observe the property of the system,” Barrett said.

He describes the ideal scenario, which involves not only collaboration, but also flexibility and context. “I think a successful conversation around non-functional requirements would lead to a development environment and a team situation where you could go up to any one developer, in an ideal world, and give them alternatives for an implementation and they would know which one was right, because they understand the non-functional requirements.”

Conversely, when NFRs are not widely understood, the IT decision-making process can become chaotic. In such an environment, when faced with a requirement for a new feature, every developer on a team may have a different take on the best way forward.

“Ten or fifteen different ideas come up and are all readily discussed and they have very different or wildly different complexity profiles, throughput characteristics or other second order effects,” Barrett said. “And there is no way to decide, because no one understands what the non-functional requirements are. No one has got a clue. So everything is done ad hoc, inconsistently.”

Barrett said that in any given project, there will always be those who understand the importance of non-functional requirements, but they are not necessarily the people who make decisions.

Still, he believes the level of understanding is improving.

“I think it is getting better. People have seen these projects go wrong enough times that now they understand that it’s probably due to non-functional requirements where most of the disappointment lies. Because I think that, actually, within the financial services industry, we are not that bad at delivering software. It may take a long time and it may be expensive, but when you get to the end of it, it tends to do what it was supposed to do. The problem is that perhaps all the things it was supposed to do weren’t specified - and that’s where the pain points come from.”

NFR pitfalls - A top ten list

Creating the conditions for business-boosting NFRs is not just about what business owners and technologists do. Often it is about what not to do. Based on the views from our experts in the field, we’ve created a top ten list of pitfalls to avoid.

- Adopting a hierarchical rather than collaborative approach to setting NFRs
- Considering NFRs late in the process (e.g. after architecture has been decided)
- Focusing on time to market above all else
- Demanding only concrete NFR numbers rather than business context
- Setting unrealistic or prohibitively expensive targets
- Not allowing for honest back-and-forth discussion between IT and business
- Calling for open-ended NFRs such as ‘as fast as possible’
- ‘Gold-plating’ a feature unnecessarily, at the expense of other factors
- Not taking due account of a business model (e.g. where is the pain?)
- Focusing exclusively on the target rather than the metrics and the trends

For more information on the companies mentioned in this article, visit:
http://weareadaptive.com
http://thetestpeople.com
http://www.thetradingmesh.com

Writing and additional interviewing by The Realization Group, Associate Editor, Adam Cox.
Financial Markets Insights from The Realization Group, is a series of interviews with thought leaders in financial and capital markets. The purpose of the series is to provide exclusive insights into industry developments, through in-depth conversations with C-level executives and key experts from banks, exchanges, vendors and other firms within the financial markets ecosystem. For more information, please visit www.financialmarketsinsights.com

Other topics in the series:
Market data at the speed of light
Addressing the Challenges of Post-G20 interest rate hedging
Fixing fixed income - taking a leaf out of social media’s playbook
The Reactive Manifesto in financial markets - trend or fad?
FPGA - Beyond market data
Bridging the DevOps skills gap
Performance considerations in FX trading platforms
The four pillars of sales productivity at global investment banks

The Realization Group is a full service marketing and business development services company specialising in the capital markets. Our team contains industry practitioners from both the trading and post trade disciplines and we have expertise equally in the on-exchange and OTC trading environments. We apply our comprehensive set of marketing programs and wide-ranging media and business networks to complement the business development requirements of our client organisations.