



The “human-to-technology” language challenge

Translating our imagination into working code

Greg Smith, Mandeep Dhillon, Laurie Guillodo, Xabier Ormaechea, Carl Bate

Within every business, software can potentially drive a significant increase in enterprise value, whether by seizing new revenue opportunities, fundamentally transforming the organization, or driving efficiencies to compete effectively against ever-evolving rivals. However, translating ideas into working software is not easy. There are notable limitations on how people and organizations “talk to” technology. This is a significant hurdle for organizations in achieving their desired business outcomes. In this article, we look at typical business-to-technology language problems, and show how to overcome these through new ideas based on our recent experience and the techniques used by digital leaders.

It is often said that within every business is a technology business, with software central to enterprise success. However, translating ideas into working software is not easy. Looking at typical business-to-technology language problems, this article shows how to overcome them through new ideas and techniques used by digital leaders.



Innovation through software

It is often said that within every business is a technology business. As the digital revolution gains ever more momentum, organizations increasingly turn to this internal technology business – for instance, when they seek to increase revenue through new or diversified products, drive down costs through optimization and automation, or figure out how to deal with a new or revitalized competitor.

Current rankings of the world’s largest companies are dominated by organizations whose principal engines of value creation are the software platforms that underpin them. Innovation increasingly resides within software, rather than physical engineering or more traditional product R&D. Whether it is pharma, automotive or telco, more and more innovation is generated and simulated within software before being made physical. Essentially, the ability to solve a problem or realize an opportunity fully within software is what will increasingly determine enterprise winners and losers.

Box 1: Medtech sector

The medtech sector demonstrates how software unlocks enterprise value. While initially created to develop and manufacture medical devices, medtech companies now increasingly focus on capturing data in order to generate valuable insight and create new revenue streams. Devices become enablers to harvest data, unlocking significant additional value. For example, iRhythm articulates its unique selling point around data (e.g., “more than 150 million hours of annotated ECG data”) and “transparent insight”, with little emphasis on its medical device itself¹. The success of this approach is reflected in its 2016 IPO, in which it was valued at 65% above its estimate².

Unfortunately, while the ability to master software is critical to future success, the barriers to achieving this for most organizations are high. Twenty years of outsourcing, offshoring and “off-the-shelf” enterprise IT solutions have left organizations with little or no internal capability and understanding of how to craft software and harness it to achieve business goals. This is being exacerbated by two trends:

- 1) the addition of digital software on top of legacy applications
- 2) the desire to move to software-as-a-service (SaaS)

So where should an organization start?

Language is fundamentally a way of transferring ideas from one person to another person or group, and the “language of business” has, for centuries, required us to understand multiple dialects specific to the organization and its operations. This could be the requirement for all senior executives to understand the “language of finance”, or for employees to learn industry- and company-specific terms.

1. irhythmtech.com

2. MobiHealthNews.com, [iRhythm goes public with \\$107M IPO, October 2016](http://MobiHealthNews.com)

With the rise of the software-enabled business, the language challenge now extends beyond the complexities of human-to-human communication and requires everyone to be bilingual. As well as being able to communicate effectively with other people, we all now need to be able to communicate effectively with technology. If we think of the many ways in a typical business day that we still fail to communicate clearly between ourselves, this is significantly amplified when we consider human-to-technology communication.

Limitations in how humans “talk to” technology ultimately reduce the value of desired outcomes, and are a key issue for organizations that are trying to leverage “digital” to transform their companies. While there is no shortage of best management theory and practice on the subject of human-to-human communications in businesses, perhaps surprisingly, there is much less understood about the equally important subject of human-to-technology communications. In this article, we would like to focus on four aspects of human-to-technology communication where we see common pitfalls and challenges:

1. User requirements – defining desired outcomes
2. Technology requirements – defining the technology architecture to prevent limitations on desired outcomes
3. Coding – writing code to achieve a working software product
4. Software product and adoption – getting users and customers to use the final product (which may include changes in their behaviors)

In combination, these factors drive businesses to waste precious resources, time and investment, as well as organizational commitment.

1. User requirements

When capturing user demand, organizations can be overly deterministic. For example, enterprise solutions such as ERP and CRM force organizations to follow the processes set out in their software, leading to a number of issues, including:

- The framework applied can feel restrictive to business users (and to technical project managers)
- The framework can be focused on describing requirements that the tool can provide – rather than the actual needs and desired outcomes of the business
- The language used to describe requirements can be overly precise

We need a language that captures the richness of demand, yet we use one that is prescriptive and artificially limited by the constraints of supply. By completing the processes set out in enterprise solutions, we fool ourselves into thinking that success will follow. Yet, once configured, the resulting solution does not meet business expectations, burdening the organization with software that does not deliver the intended benefits.

2. Technology requirements

It is easy to believe that if a language is not descriptive enough, it can be a limiting factor. Yet, this contradicts with the underlying technology that unleashed the internet. Representational State Transfer (REST) is a technology architecture pattern that helped the internet become what it is today. Its language is based on just four verbs: GET, POST, PUT and DELETE. RESTful web services provide interoperability between different computer systems – with these four verbs, the only limitation to leveraging online computer systems is one’s imagination.

The simplicity of REST directly contrasts with the complexity of Simple Object Access Protocol (SOAP) – another computing language originally defined in 1998 that allows web services to “talk to” computer systems on the internet. SOAP has a much richer descriptive framework, which is also more complex, and therefore restrictive.

At the time, the thought was that the richer and more complex the language descriptor, the more one could accomplish, yet REST has proven this is not the case. The complex framework that SOAP uses to connect web services ultimately drives an equally complex

and constraining manual/bible of enterprise tools.

We therefore need a language that unleashes imagination and creativity, yet many enterprise tools are built on technology protocols that limit and constrain. By falling into the trap that more technology descriptors help codify requirements, we actually limit the business value through multiple, complex and constraining protocols.

3. Coding

Coding today has a fundamental limitation – the number of people that can code. This scarcity of resources pushes up salaries and therefore means that organizations face budget issues when it comes to employing programmers. How do we, therefore, allow more people to create software, so that it becomes a process, rather than requiring them to learn to code?

To do this, we need a language that removes the barriers between ideas and execution as working code. Yet, we have a human-to-technology language taxonomy that, despite best efforts, has multiple barriers to entry. The end result is that we limit the number of people with great ideas who can quickly create business value.

4. Software product and adoption

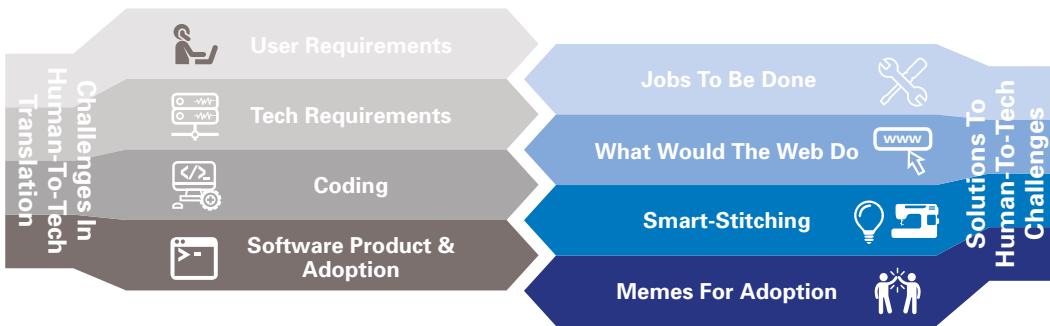
Adoption of new technologies within organizations is often met with passive resistance or “tissue rejection”. A recent Arthur D. Little article³ identified some common pitfalls, which are compounded by the issues described above. For example, many organizations underestimate behavioral issues – failing to adequately plan how employees will adopt the tool – and mismanage communication, which impact adoption.

We need a language that provides meaning by reflecting real-world complexity, and is simple to understand. Yet, we often neglect this and its impact on new technologies.

3. [See “Tackling the Digital Hype”, Prism 2016](#)

How we “talk-to” technology to translate our ideas to working code

Overcoming this challenge is not easy, but the following framework provides a blueprint for improved human-technology communications.



1. Jobs To Be Done

Enterprise software processes can limit the innovation potential of organizations. Therefore, when selecting or building new tools, it is important to deploy design-thinking concepts to help capture requirements that can test outcomes in a meaningful way:

1. **User stories** – this captures actor, action and benefit, and is always expressed in a specific format (“as an <actor>, I can do <action>, so that I get <benefit>”). Through workshops, this tool helps business users define what they want. The only limitation of this method is the imagination of the business user, which can be constrained by how “they’ve always done it”; this is why it is vital to capture the benefit. By understanding the action and benefit in conjunction, one can determine the difference, and test and validate different scenarios with end users.
2. **Personas** – this captures examples of users and their “pain” and “gain” points – i.e., what makes their lives difficult today, and what would remedy their issues. Personas also capture how a user may use a technology (e.g., frequency, location, device), which has an impact on overall technology requirements.

3. **Jobs To Be Done** – a key component of personas and user stories is to define the ultimate desired outcome. These are expressed as “Jobs To Be Done” (JTBD), as explained by the JTBD framework developed by Clayton Christensen⁴ and Anthony Ulwick⁵. This is the core desired action and outcome – and removes the impediment of how people have worked before to open up new ways to tackle old problems.

Arthur D. Little recently supported a major manufacturing company looking to select new tools to manage its innovation. Its choice was between traditional enterprise tools (which were costly and complex, and would take time to deliver value), a new software-as-a-service tool (which may not have provided the required functionality and was perceived to be risky) and building its own. By applying JTBD, Arthur D. Little was able to quickly test >40 tools in the market, and provide a real choice, articulating the pros and cons based on user stories and JTBD (supported by a traditional three - and five-year business case).

2. What Would the Web Do (WWWD)

REST is an architecture pattern that has underpinned and propelled the internet to where it is today. Other similar technology patterns have been adopted to help drive and deliver innovative digital solutions, and using these Digital Design Principles helps deliver robust, performant and highly extensible solutions to meet business needs. The ultimate purpose is to deliver technology architecture patterns that are open and flexible, which do not limit desired outcomes. User stories and JTBD are useful to help test these requirements.

4. [Clayton Christensen, Scott Cook and Taddy Hall, “Marketing Malpractice: The Cause and the Cure”, Harvard Business Review, December 2005](#)

5. [Anthony W. Ulwick, “What Customers Want: Using Outcome-Driven Innovation to Create Breakthrough Products and Services”, McGraw-Hill, 2005](#)

Some examples of Digital Design Principles:

- *Design for Uncertainty* – build for flexibility, and enable strategic agility
- *Leading Edge, not Bleeding Edge* – ensure tools are first class and proven, but also appropriate for your needs
- *Automated, Scalable and Agile* – build for automation by default, and ensure this is reflected in your business case
- *“Openable” Data and Interfaces* – fully accessible, one version of the truth.

An airline company that wanted to provide better situational awareness to its business-crisis teams followed these principles. By using RESTful interfaces to extract data from existing legacy enterprise tools, combining it, and then providing a dashboard view, it helped employees on the ground make decisions based on new and emerging insight. A significant benefit was that it allowed teams to hold their nerve and not cancel flights, as they could see from the dashboard that the incident was improving.

3. Smart-stitching

Traditionally, organizations looking to acquire software had two fundamental approaches – build or buy. Unfortunately, both of these approaches have proven to be problematic, leading to the poor track record of IT project delivery and frustrated business leaders. The main challenges for a 100% built solution are normally time and cost, and the shortage of skills within an organization to deliver – and, just as importantly, to manage – a bespoke software development process. Unfortunately, bought packages have different but equally debilitating problems. The change required to match business processes to standard configurations and flows within software packages has, all too often, proven impossible for businesses to adapt to in anything other than trivial or genuinely commoditized domains.

Fortunately, best practice among technology innovators and software-driven businesses has opened up a third approach that we describe as “smart-stitching”. In this pattern, an end-to-end business capability can be split into elements that can be delivered using standard software platforms, which are increasingly open source,

cloud based and delivered as a service. When the analysis of what is needed is carried out, the elements that are truly bespoke to a solution or organization are typically a small percentage of the total footprint, and custom development of these components becomes a manageable and time-efficient exercise. The commodity and custom elements can then be “stitched” together into an end-to-end solution.

Another significant benefit of the smart-stitching approach is the ability to exploit legacy technologies using the same techniques. Once legacy platforms are looked on as an asset to be harnessed and exploited, rather than an obstacle requiring wholesale replacement, the footprint of the required solution often shrinks dramatically. Often the question asked by an organization needs to be reframed to: “What is the minimum change we need to make to achieve new capabilities?”

Critically, when considering a “smart-stitch” approach, companies should always put “design before technology” to ensure that the solution is wrapped with a user experience (UX) that is intuitive and designed to drive adoption. This ensures that the diversity of elements comprising the overall solution (legacy, commodity and custom) is hidden from the customer or user, who has an experience built around their needs.

A large entertainment and gaming company used smart-stitching to deliver an innovative, new customer point-of-sale omnichannel experience. This involved taking existing and new technologies, and linking them to enable automation of staff tasks, in order to provide a better customer experience, and to help them better meet emerging government regulations.

4. Memes for adoption

The word “meme” conjures a mental image of a joke that is spread online through social platforms, but it is also a powerful business tool to pass on an idea or new information over two speeds: quickly through viral adoption, and slowly by embedding values and culture. Indeed, Arthur Dehon Little’s response of, “Who says it can’t be done?” when asked whether it was possible to create a silk purse out of a pig’s ear, is an excellent example – one that has inspired both consultants and clients since 1886.

Memes act by delivering a simple message. The purpose of that message can be two-fold: to deliver a quantum of information to explain and enact a change or desired outcome, and/or to help embed adoption of that desired outcome into the DNA of an organization's ways of working and culture. The first must happen very quickly, almost intuitively, to the receiver of the message. The second happens over time, as a feature/by-product of the ways of working and culture, through peer-effect. Both require a meme to have similar characteristics: simple, short, and salient. Another important characteristic is that the meme explains the desired outcome.

There are numerous historic examples of companies applying memes to help deliver internal change (e.g., "Change before you have to" – Jack Welch, General Electric; and "Obsess over customers" – Jeff Bezos, Amazon). Emerging digital disruptors are particularly good at embedding memes reflecting their ways of working. Netflix recently updated its 100+-slide pack⁶ into a 10-page prose⁷. The two documents contain a series of powerful memes (e.g., "People over process", "Context, not control", "Highly aligned, loosely coupled") – so powerful that they have actually spread way beyond the company's borders: one version of the initial deck has more than 16 million views on SlideShare.

Arthur D. Little has used memes with its clients to help drive adoption of new ideas and technologies – for example, the meme, "show me, not tell me", which is fully aligned to agile⁸ ways of working. Our teams use this to create a "slice" of the working solution – not throwaway code – to show the art of the possible through a working proof of concept. Critically, it produces a tangible product in weeks (not months or years) that can be tried and tested by business and technology stakeholders, or customers, for early feedback, iteration and buy-in.

6. slideshare.net/reed2001/culture-1798664

7. <https://jobs.netflix.com/culture/>

8. Agile Manifesto: <http://agilemanifesto.org/>

Insight for the Executive

Increasingly, companies will win or lose based on the value they can generate in the software layer. This requires executives to help their organizations improve their ability to “talk to” technology. However, traditional approaches are constrained, by either restrictive language and frameworks, closed technology architecture patterns, or not thoroughly considering adoption. There are new ways of thinking and working that can help improve the effectiveness of human-to-technology knowledge transfer, which are summarized by:

1. Jobs To Be Done
2. What Would the Web Do?
3. Smart-stitching
4. Memes for adoption

Wikipedia’s definition of technology is: “... the collection of techniques, skills, methods and processes used in the production of goods or services...” In this sense, the language we use to talk to machines, to conceive and codify software, is a type of technology itself. We therefore recommend that executives look at how their organizations can understand and get better at the technology of language, if they want to drive success.

Greg Smith

is a Partner in the London office of Arthur D. Little, and a member of the Digital Transformation Practice.

Mandeep Dhillon

is a Manager in the London office of Arthur D. Little, and a member of the Digital Transformation Practice.

Laurie Guillodo

is a Manager in the London office of Arthur D. Little, and a member of the Digital Transformation Practice.

Xabier Ormaechea

is a Consultant in the London office of Arthur D. Little, and a member of the Digital Transformation Practice.

Carl Bate

is a Partner in the London office of Arthur D Little, and a member of the Digital Transformation Practice.