OLIVER WYMAN

€15 TO 30 BILLION
COUNTING THE COST OF INEFFICIENT SOFTWARE R&D

REAPING THE FULL VALUE FROM SOFTWARE BUSINESSES

The global software market has become huge in scale and is growing rapidly. Many companies are investing in bold agile and DevOps programs, yet at a time where firms are becoming even more cost conscious, in many cases inefficient and outdated practices are diluting the success of their software development.

According to Oliver Wyman's analysis of the European Union's industrial research and development (R&D) data, almost €100 billion was invested in software R&D by the top 2,500 companies during 2018. Furthermore, this trend is set to continue, with five percent compound annual growth (CAGR) forecast until the middle of this century (2017 to 2050).

Software is often one part of the bigger picture of integrated products and services. Seamless integration is a key issue for all those involved in the development of products and services, and not just for software R&D. This places enormous pressure on engineering teams to ensure that their software development processes are effective. Achieving this demands the alignment of software development practices across the entire organization, not just within R&D or the individual development team.

This generates big questions for those who are responsible for developing software or who own software companies, such as private equity funds:

- How do we build a sustainable and profitable software business?
- Which full suite of tools and techniques do we need to support successful product development?
- How do we ensure that our development processes are fully integrated as well as effective?

To explore these questions, Oliver Wyman conducted a study of the current state of software development. Based on this research, we examined the issues currently facing the industry and have outlined guiding principles for successful software development in today's environment.

MASSIVE SOFTWARE R&D INVESTMENTS

Software is one of the most R&D-intensive industries: On average, leading software companies invest roughly 10–15 percent of their revenue in R&D.

Despite the already high spend on software R&D, the scale of investment is growing rapidly. During 2017–18, the industry saw an average increase in software R&D spend of 26 percent. Just as for the industry as a whole, this R&D investment is concentrated in software

companies based in the United States; these companies account for 73 percent of the global software R&D spend. However, the rate of investment growth is even higher elsewhere in the world, topping out at 30 percent in the Asia-Pacific (APAC) region (see Exhibit 1).

Exhibit 1: R&D expenses as a percentage of revenues for software development companies, 2018

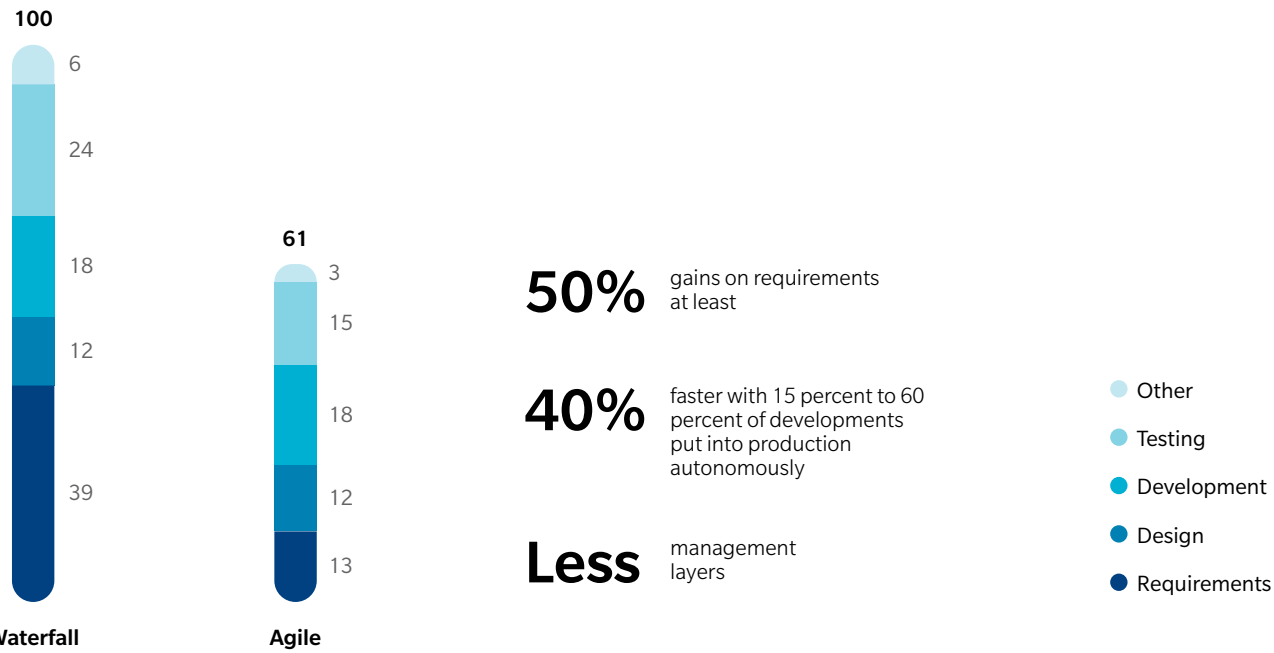
	AMERICAS	EUROPE, MIDDLE EAST, AND AFRICA	ASIA PACIFIC
Intensity	10.4 percent	15.8 percent	4.6 percent
Investments	€69 billion	€10 billion	€15 billion
Growth (18 versus 17)	+20 percent	+26 percent	+30 percent

Source: 2018 EU Industrial R&D Investment Scoreboard, Oliver Wyman analysis

Software development is challenging on many dimensions, including throughput, cost, and quality. Software errors range from the fairly minor, which cause no more than frustration for the user, to highly serious and even fatal. This inefficiency costs companies dearly, damages their reputations, and is symptomatic of deeper problems.

When looking at the EU's data based on the top 2,500 companies, Oliver Wyman estimates that almost €100 billion was spent in 2018 on software R&D and engineering. Yet a large part of this investment is wasted through inefficiencies that could be significantly reduced by using best practice software development techniques (see Exhibit 2).

Exhibit 2: External benchmark on application development cost.
Waterfall methodology versus agile, base 100 for waterfall methodology



Source: Oliver Wyman analysis

Such techniques can cover different areas of the business and allow companies to boost efficiency and optimize costs. These techniques encompass:

- Using agile methods to better align expectations and clarify requirements.
- Using the cloud to distribute and maintain products.
- Improving the level of quality through an automated test-driven approach. For example, Stripe carried out a study in which data was collected from 1,000 developers and 1,000 software executives (*The Developer Coefficient*, 2018). The study revealed that, on average, software developers were spending half their week on product maintenance and fixing bugs, with too much of their time dedicated to correcting poor code. As a result, capacity was being wasted instead of being used to create value.

Overall, Oliver Wyman estimates that not making full use of best practices in software development represents lost value of approximately €15 to €30 billion.

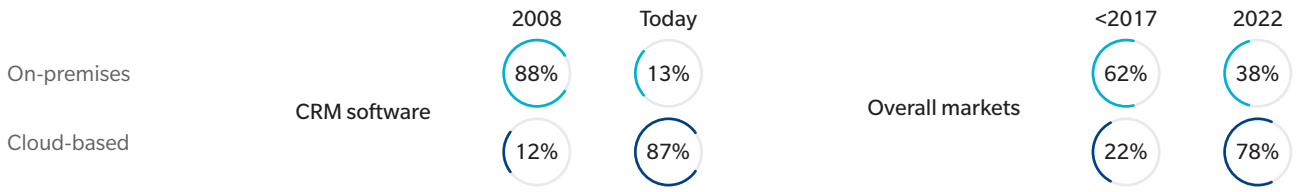
THE CHALLENGES FOR SOFTWARE BUSINESSES

Software companies that have existed for more than 10 to 15 years, and which are based on the historical "on-premises" software model, face a major transformation (see Exhibit 3). More and more software products are cloud-native or even based on software-as-a-service (SaaS, such as enterprise resource planning (ERP), customer relationship management (CRM), finance management, or travel management), usually with a pay-per-use structure rather upfront payment. This affects companies' investment plans, technology issues, and day-to-day working methods.

In addition, software development is often part of the bigger picture of products and services. This requires companies to make use of the entire spectrum of tools and techniques available to them, from managing the interface with the customer to ensuring the highest quality. While many are familiar with "agile" and "DevOps" (software development and IT operations), these approaches are only part of the whole picture.

Our study suggests that very few organizations are approaching development satisfactorily; they are placing more emphasis on one or two aspects and neglecting others. We explore these issues in greater detail below.

Exhibit 3: The changing software environment



Source: IBM

Source: IDG, CIO Tech Poll

Exhibit 4: Sample best practices
Abstract from Oliver Wyman's maturity model

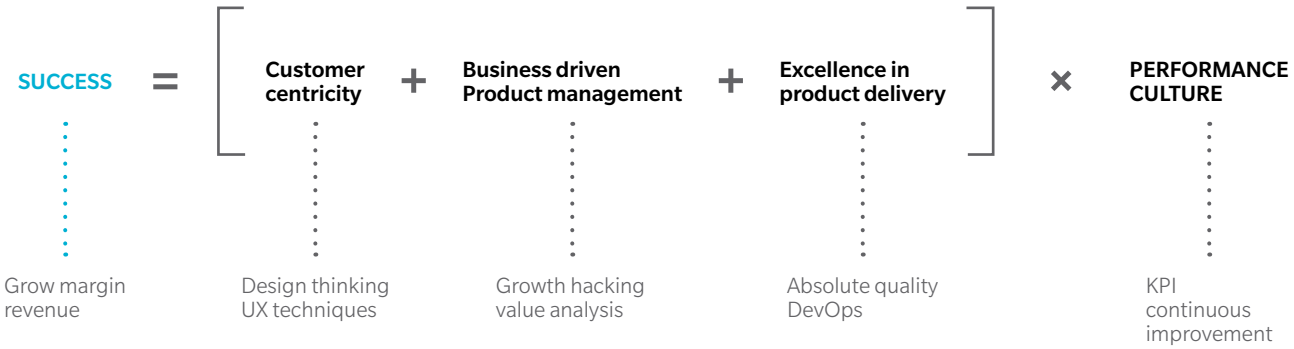
	Dimensions	Value	Leaders
Product management	Customer centricity	**+++**	• Continuous feedback loop with customers • UX / UI and design thinking techniques with user test by persona to support roadmap • Generalized usage analytics • Positive NPS (40 to 60 in B2B ; >100 in B2C) • Clear product strategy • Growth hacking
	Value analysis	++	
	Portfolio optimization	++	
	Growth hacking	+	
Technology	Cloud	**+++**	• Cloud-native or SAAS products >80% • Technical debt cleaned continuously • Framework based development • All libraries maintained • Full API and micro services • Clear plan to use open source and third party services • API portals in place
	Technical debt	++	
	Reuse of components	++	
	Architecture	++	
	Open source	+	
Software engineering	Quality	**+++**	• €700 to 1000 K / R&D FTE • Excellence on quality. Less than 1 critical event / year / product • > 90% automated tests, including chaos testing on cloud • Auto-healing, auto scaling in place • Agile and DevOps at scale with full KPI • >1 release per week, with continuous deployment and feature flipping in production, • Developer NPE engagement > 50
	Talent	**+++**	
	Ways of working	++	
	Tooling	++	
	Sourcing and IP	+	
	Organization and governance	+	
Performance management	Metrics	**+++**	• Consistent metrics across the firm supported by tools • Compensation indexed on performance • Systematic improvement loops
	Continuous improvement	++	

Source: Oliver Wyman Analysis

THE EQUATION OF SUCCESS IN SOFTWARE DEVELOPMENT

Leading companies, as shown in Exhibit 4, build successful software businesses by combining four elements:

1. Successful product management.
2. The best technologies.
3. Excellence in software engineering.
4. The right performance management culture.



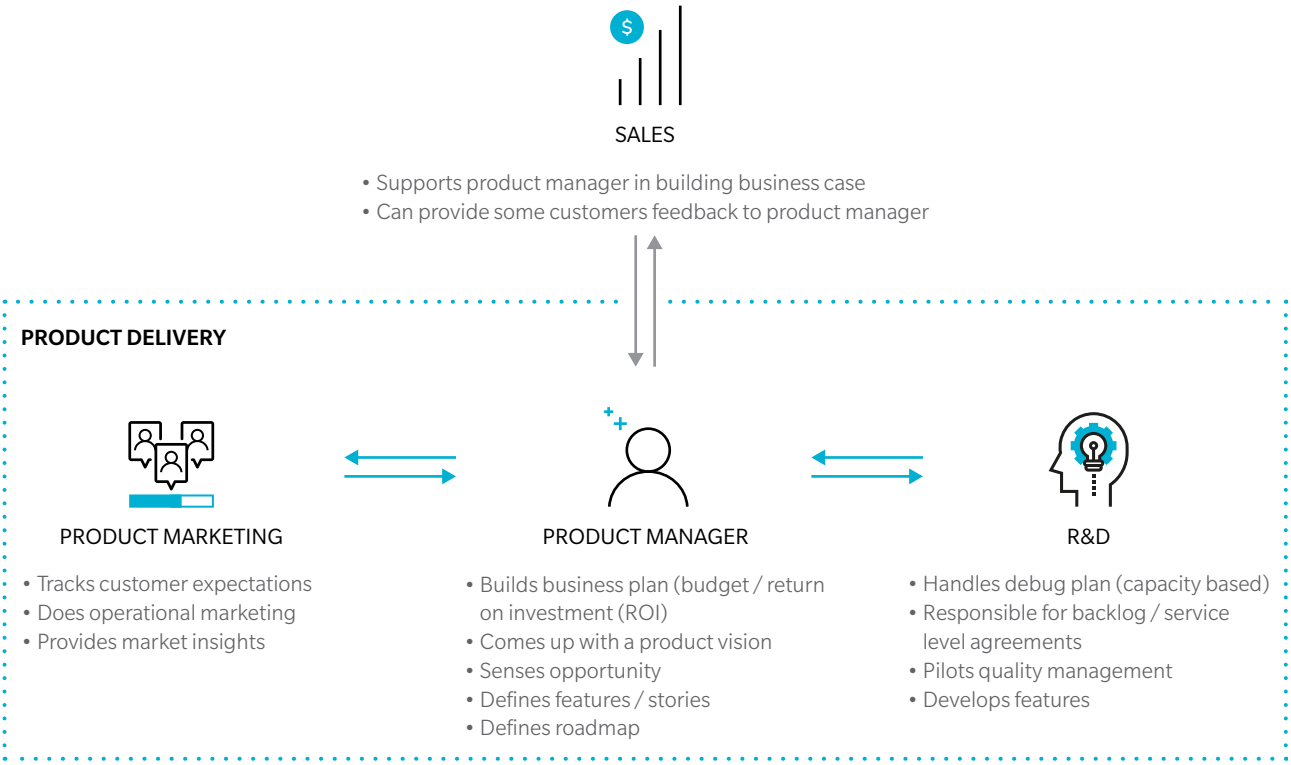
1. CREATING SUCCESSFUL product management

Product managers play a pivotal role in successful software development. Not only do they liaise with sales, R&D, and marketing but, above all, they also shape the product development roadmap. This ensures they have a focus not just on what the product does but also on how it will be used and who will use it. Product managers also take a lead in growth hacking to ensure the rapid acquisition of customers through targeted actions that have a clear business rationale.

Product managers push their customers to use the most recent version of their product. At the same time, they ensure older versions continue to be supported at a low or minimum level of maintenance while pushing for their decommissioning. A useful rule of thumb is that when a product is not SaaS-based, and therefore we can't be certain that the latest version is available to the customer, no more than two versions of the product should be running at the same time.

Most importantly, it is essential to ensure that migration to the new version is pain-free for the customer. Prior to the cut-off date for supporting an older version, companies need to communicate this deadline (and the product roadmap as a whole) to the customer – as well as to the product support team (see Exhibit 5).

Exhibit 5: The role of product managers in communicating the software development process



Source: Oliver Wyman analysis

Customer centricity is the second element of successful product management. Software development and integration starts and ends with the customer. No product can be deemed successful unless the customer is fully satisfied. It is therefore essential to ensure that customer involvement is central to the development process. A truly customer-centric process needs to be dynamic and to incorporate at least three ingredients: strong and frequent communication with customers, constant feedback from the market, and prototype adaptation. This relentless focus on customer experience is most frequently achieved by creating boards of users that provide constant feedback during development.

A customer-centric approach should be: implemented in the development of each product component; supported by analysis of the value to the customer; and combined with demand management analysis. Best-in-class companies seek to develop deeper insights about their users by involving customers not only in the design of the initial product but also in the process of continuous improvement. This ensures that software is never developed in isolation but is always treated as an integral part of the final product or service.

Customer-centricity has been shown to have a real and meaningful impact on the development process. In leading technology companies, the development of some 25 percent of features is stopped following tests with customers. As a consequence, while certain features are never implemented, the product manager can be sure the ones that are produced bring real customer value.

2. USING THE BEST TECHNOLOGIES

Leverage SaaS and the cloud. Cloud technologies now offer scalability, availability, elasticity, and an always-up-to-date infrastructure at a far better price point than previous approaches. Software companies can explore several ways to leverage cloud technologies:

- Use cloud infrastructure (such as infrastructure-as-a-service (IaaS) and platform-as-a-service (PaaS)) for software development activities (for example, load testing or non-regression testing in non-production environments) and for marketing (for instance, to create dedicated demo environments to support sales).
- Use cloud technologies to reduce software delivery and support costs (for instance, by packaging software assets as a set of Docker containers), which in turn reduces installation costs, incompatibilities with on-premises setups, and so on.
- Use an external application programming interface (API) to extend current software capabilities (mashup extensions like localization services, payments services, and so on) or specific cloud services (such as cloud-based storage, artificial intelligence (AI), and machine learning services).
- Provide the solution as PaaS or IaaS and manage its runtime on behalf of customers.

Migrating a software package as a cloud or SaaS solution requires a level of sophistication in DevOps and multi-tenancy that only a mature player can achieve.

Leverage shared components across software. Optimize costs by reusing components and leveraging open-source features. Leading developers reuse components on a regular basis, and more than 80 percent of each new product is comprised of components from other pieces of software. This approach greatly reduces costs and simplifies maintenance. But it also requires continuous and strict library management.

Not all software can be reused but certain elements can be designed from the outset for potential reuse. The most common reusable components relate to:

- Infrastructure, for example tenant management, data access, and containers.
- Design systems, such as user experience or user interface design, and design patterns.
- Standard business issues, like identity management, API management, workflow management, and search functions.

These components are usually accessible via coding frameworks and libraries and, because they evolve over time, they need to be updated.

It is therefore key to think about the product portfolio in terms of a modular platform. This means avoiding building a monolithic solution. Instead, assemble external components that are proven and need minimal customization, and develop a clearly defined management process of the dependencies with external frameworks, libraries, and components to facilitate future evolution.

Looking ahead, the next wave of automation will focus on combining AI capabilities with cross-platform compatibility. In this environment, software development will no longer be thought of as a stand-alone exercise. Development organizations will need to ensure they have the abilities necessary for integrating ongoing developments from third-party

developers into their own proprietary development programs. This challenge is probably the next level in software integration.

Leading software companies are now preparing for the next development in distributed computing – that of edge devices. These can be classified as any device that sits outside of the data center. In this model, computer and storage resources are placed at the "edge" of the internet, near to mobile devices or sensors (security cameras or self-driving cars, for example). This trend will produce further challenges, such as being able to manage the impact of local developments, the need to perform immediate diagnosis at the edge, and the delivery of consistent performance in a highly distributed environment.

3. ENSURING EXCELLENCE IN PRODUCT DELIVERY

Quality is the bedrock. Product excellence must become the standard for the entire software organization. To support this, the full range of tools and techniques should be introduced to enforce best practice in quality control. These include those for ensuring clarity in requirements management, using tools, and setting minimum standards. The use of these tools and techniques needs to be supported by a clear stage containment process, automated testing, and routines to investigate problems that arise outside of the normal quality control practice. A strong quality-oriented culture is required to underpin the entire approach, one that is embedded throughout the organization.

Build flexibility through continuous delivery. DevOps best practices enable high-performing organizations to make changes very rapidly and have a high impact on productivity. In one example, the code editor capacity of a major software developer was increased by 60 percent once DevOps were implemented at scale.

Build an agile organization. In best practice companies, lead teams are organized around features. They are able to leverage state-of-the-art infrastructure, tools, and development frameworks. Development is carried out through a series of sprints, and each sprint lasts one week. The final day of the sprint is dedicated to quality (code review or tests, for example).

Secure the right talent. Obtaining the right human resources is vital. Highly skilled developers – competent in modern technologies and code languages – are essential for managing innovation and product quality. While software development frameworks are powerful ways to organize the work, they can achieve little or nothing without a great team to make use of them. Other technical skills are required, including in data structures, an ability to pay close attention to detail, and a basic mastery of testing, networking, and security. Technical skills should be complemented by soft skills, especially logical thinking and problem-solving and interpersonal skills.

The right human resources need to be in place to ensure that all the developer's products achieve the quality required by customers, and that the software is free of bugs and is as simple as possible to maintain. Because leading developers know just how valuable the right people are, they not only secure the necessary skills but also structure individual career paths to retain them.

4. BUILDING THE RIGHT PERFORMANCE MANAGEMENT CULTURE

Beyond delivery and execution, creating transparency through targeted key performance indicators (KPIs) is key. Leading developers steer all their activity with the use of just a few critical KPIs. These metrics are cascaded across the enterprise, at every level and to every function, thereby ensuring consistency in goals and objectives throughout the entire organization. This alignment underpins the drive for meeting customer requirements and for ensuring bug-free, successfully integrated software. The staff are also incentivized by positive outcomes and all teams strive for continuous improvement.

Examples of typically observed KPIs are R&D contributions, velocity, number of bugs, SaaS availability, customer satisfaction, and talent retention.

BARRIERS TO RAPID CHANGE

The software industry is undergoing a massive transformation. SaaS is becoming dominant and is well on the road to replacing client-based software in the near future. But it is not easy for legacy vendors to embrace a holistic transformation as there are still many barriers to overcome.

These companies will need to:

- Pivot their business model from on-premises to SaaS. It will impact customers willingness to pay and their finances moving from CAPEX to OPEX models. The transition must be explained (for example, to shareholders) and careful communication about finance will be of critical importance during the transformation: it could affect companies' valuation.
- Migrate the installed customer base to SaaS-based versions. The workforce will have to be reskilled or upskilled so they can master new technologies and radically change ways of working, including mastering agile and Devops and acquiring the skills and experience to operate at scale multiple customers in production.
- Manage resistance to change, as most of the people implementing the transformation will be the same people who developed the legacy software and practices. The whole DNA of the company may have to be reviewed.
- Manage business as usual in parallel of the transformation.

THE WAY FORWARD

Many independent software vendors and larger software companies are far from excelling, and the case for change is clear: Succeeding will mean increasing the revenue streams but, above all, it will mean securing the sustainability of a profitable business in the medium to long term.

To get to that point, a major transformation is needed. A successful process would, typically, highlight clear actions and milestones against product management, technology, software engineering, and performance management.

1. Product management: Review the product portfolio and be clear on the strategy for each product. Set up mechanisms to continuously capture the customer voice and to prioritize products and features.
2. Technology: Identify technologies of the future and build a plan to bridge gaps and to migrate products to modern technologies. Be clear about the level of technical debt and accuracy of libraries.
3. Software engineering: Build a plan to move to Agile and DevOps at scale and a plan to upskill the workforce. To excel, pay close attention to quality in these plans.
4. Performance management: Set up basic metrics across the whole organization and develop a strong performance culture.

It is important that these streams are launched and maintained concurrently, and the initial challenge will be to secure the first levels of sustainability. However, managing software development in this way will give your firm the freedom to make the right choices over the long term, choices that are required to ensure that less value is lost through inefficient and outdated R&D practices.

AUTHORS
Marc Boilard
Xavier Boileau
Charles de Pommerol
Mohssen Toumi

Oliver Wyman is a global leader in management consulting that combines deep industry knowledge with specialized expertise in strategy, operations, risk management, and organization transformation.

For more information please contact the marketing department by email at info-FS@oliverwyman.com or by phone at one of the following locations:

AMERICAS	EMEA	ASIA PACIFIC
+1 212 541 8100	+44 20 7333 8333	+65 6510 9700

www.oliverwyman.com

OLIVER WYMAN