PART ONE

e-Business and the Changing Role of Application Development

ONE

INTRODUCTION

Much has been written about the impact of computer technology on all of our lives. As the availability and use of computers has increased, so have the valuable and innovative ways in which computer technologies have been employed to improve every aspect of the way we live, conduct business, and communicate with each other.

While this background is well-known and well-documented, a key aspect of this story is much less frequently highlighted; the challenges this brings to organizations tasked with developing, deploying, maintaining, and evolving software-intensive solutions designed to make use of this technology. In this chapter we examine the challenges software development organizations are facing in the context of current computer technology developments. This leads us to consider the primary business drivers motivating those organizations relying on computer-based technologies to carry out their activities. In this way, we are able to distill the primary requirements placed on software development organizations and the techniques they employ to develop, deploy, maintain, and evolve the systems they build.

1.1 MOTIVATION

We all frequently interact with software-intensive systems. The impact of their availability (and failure) is of everyday concern to large parts of the world's population. However, what is often ignored is that the software industry involved with the development and maintenance of software systems is increasingly becoming a strategic aspect of many countries' economies. The software industry has an important and compelling impact on economic, safety, and employment aspects of the world. For example:

- In 1998 it was estimated that the worldwide market for programmer development tools was in excess of $24 billion, and is continuing to grow at a rate of over 11% per year [1].
- Computer systems are at the heart of many safety-critical systems such as aircraft control, hospital patient monitoring, and power station management. The consequences of failure of these systems are severe. For example, in 1996 the uninsured loss due to the crash of the European Ariane 5 launcher was estimated at $500 million, and was subsequently traced to a single error in the Ariane's control software [2].
- By the end of 1999 there were almost 200 million Internet users worldwide. By 2003 it is estimated that more than 500 million will be "surfing the web" [3]. This offers a huge potential market for goods and services, and a significant base of employment opportunities.

While the ubiquity and importance of software is not in doubt, and certainly the consequences of software-induced failures are well-documented, perhaps most worrisome to many of those involved in the software industry is the lack of control and predictability of many software projects. Many surveys attest to the difficulties organizations have in predicting software costs, in identifying suitable target technologies to use during software development and maintenance activities, and in gaining visibility and control throughout the software life cycle.

As a consequence, the last decade has seen the promotion of software development as an engineering discipline, encouraging the use of more rigorously defined software development methods based on sound mathematical principles and supported by a strong experience base. This has been backed up by initiatives with titles such as the "software experience factory" [4] and the software capability maturity model (CMM) [5], by software certification efforts such as ISO 9000 [6], and by a host of new software development methods, tools, and techniques. While each of these initiatives has improved our capabili-

ty to develop and maintain software, there is little indication that the problems faced by software development organizations are either reduced or less frequent.

A major reason for these ongoing problems is that the scale and complexity of software-intensive systems is constantly growing. Throughout the past three decades there has been increasing use of computer technology in a growing number of areas. We can identify at least three important reasons for this.

First, there has been a dramatic increase in readily available computing power. Computer main memories are now measured in hundreds of megabytes, secondary memories in gigabytes, and personal computer central processing unit (CPU) speeds in hundreds of millions of instructions executed per second. Furthermore, despite predictions to the contrary, new technology advances indicate there is no immediate end to the trend of increasing computing power. Moore's law (that processor speeds double or prices halve every 18 months) continues to prevail.

Second, there has been a continuing fall in the real cost of this computing power. The availability of powerful desktop computers at low cost has made computer technology available to most businesses, and to many individuals. This, coupled with advances in computer networking (including intranets and the Internet), has allowed organizations to harness the computing power they have with less concern for its physical distribution across the organization.

Third, there has been a growing acceptance that information technology is a pivotal component of many business functions. Computer technology is seen not only as a significant business enabler, it is now frequently considered an integral part of an organization's competitive advantage, significantly impacting the way they do business, compete in a crowded market, and differentiate their products and services.

1.2 SOFTWARE DEVELOPMENT CHALLENGES

This rapid increase in the power, availability, and use of computer technology has had a number of important consequences. For those responsible for provisioning and maintaining software-intensive systems, the most compelling of these are:

- the demand for computer software has not been met by software producers at the rate required (the so-called "software crisis");

- the complexity of computer software is constantly growing, both in the size of developed systems, and the intricacy of its operation to meet the extensive functional and non-functional requirements placed upon it;
- the rate of change of computer technology has increased the risk that projects with a long life cycle can have early technology decisions made obsolete by later technical advances;
- the significant investment made in developing and deploying computer technology cannot easily be thrown away when business requirements change—many organizations have mission-critical systems in place that were developed over a decade ago.

Clearly, as software-intensive systems reach this scale and complexity, the problems of software production become significant. Producing such systems within budget and on time is problem enough. Designing a system so that it can easily evolve as the operating environment changes, as user requirements are modified, and as errors (which inevitably occur in such systems) come to light, makes this an even more difficult task. Yet these are the challenges faced by any organization involved with the development, deployment, maintenance, and evolution of software-intensive systems.

1.3 THE KEY TO THE FUTURE: MANAGING COMPLEXITY AND RAPIDLY ADAPTING TO CHANGE

Organizations that deploy software-intensive systems face pressures from many sides in their efforts to improve the way they carry out their business activities. The biggest challenge they face is how to manage the complexity inherent in the systems they are deploying, while at the same time being able to rapidly adapt to change. This combination of complexity and change provides the greatest risk to successful system deployment.

To investigate this challenge further, it is useful to examine the sources of both of these key ingredients: complexity and change.

Managing Complexity

To be successful, the systems being developed and deployed must meet a variety of requirements. The complexity of any software-intensive system naturally tends to increase as these requirements grow in quantity, diversity, and difficulty. While there has always been a range of complex systems being developed, the past few years has seen a marked increase in system complexi-

ty. The source of this complexity lies in at least four classes of requirements being placed upon them.

Functional Requirements

Software-intensive systems are being deployed in a much wider set of application domains than ever before. Increasingly, computer-based solutions are at the heart of innovative technology approaches in domains as diverse as the aerospace, manufacturing, financial management, and entertainment industries. Each of these industries brings a range of new functional requirements to software developers, frequently pushing the boundaries of capabilities previously offered.

Furthermore, as technologies evolve, whole new domains of business are opening up to take advantage of them. One prime example is the impact that electronic commerce has had on the retail industry. As a result of technology improvements such as the Internet, completely new forms of trade between suppliers and users of retail goods are possible. This has resulted in significant changes to the requirements for many systems being developed for those industries. For example, order management systems must now be concerned with a variety of nontraditional means of order entry, payment, and delivery of retail goods for people making online orders over the Internet, making payment using electronic forms of cash, and allowing real-time tracking of delivery, inquiry, and order amendment.

Nonfunctional Requirements

A software-intensive system that performs the functions it was designed to do is necessary but not sufficient. It must also do so accurately, reliably, predictably, securely, and quickly to support the volume of use expected. These required attributes of a system are often referred to as the system's nonfunctional requirements. As a consequence of the ubiquity and importance attached to the operation of software-intense systems, nonfunctional requirements are becoming ever more important to the systems being developed and deployed.

Many organizations rely heavily on computer-based systems to carry out their daily activities. For such organizations there are costly consequences if those systems are unreliable, unavailable, or unusable. Hence, mission-critical applications are typically specified with stringent requirements for mean time between failures, startup times in the event of problems, number of simultaneous users, transaction throughput, and so on. Meeting these requirements typically results in a great deal of additional complexity in the provisioned solution. Approaches employed include the use of duplicate hardware and

software, fault-tolerant designs for exception handling, transaction logging and rollback to enable hot standbys, and various multilevel security schemes. Each of these approaches will add significant complexity to the solution.

Technology Requirements

In recent years there have been many improvements to the technologies available for developing and deploying software-intensive systems. Consider one aspect of this new technology, namely, the platform on which applications are deployed.

The availability of low-priced personal computers has accelerated the requirement for software-intensive systems to target distributed networks of computers as the platform on which they must execute. Whether this results in a central computer running most of the business logic accessed by remote desktop machines, or true peer-to-peer systems of interconnected machines, significant complexity is introduced to the applications to allow them to execute in such an environment. Indeed, as the move to distributed computing accelerates, these requirements become more severe, and may lead to more and more complex solutions.

As an illustration, consider the variety of target execution platform options that may be available to a typical application. One such configuration is illustrated in Figure 1.1. As a result of this architecture, any application designed to execute on it must devise some approach to handling communication among the various pieces, synchronization of the actions performed across the pieces, and graceful degradation in the event of failure of any piece.

Organizational Requirements

The organizations developing and deploying software-intensive systems are themselves distributed across organizational boundaries and, increasingly often, also distributed geographically. For any project distributed in this way the development process for software must cope with, and if possible take advantage of, this situation. Going back to the earliest days of software development projects such as the IBM/360 operating system, Fred Brooks noted that the way the development team is organized could have significant impact on the structure and complexity of the applications being developed [7]. This complexity is only likely to increase given recent moves toward distributed development teams, telecommuting, and offshore development.

External Software

An additional source of complexity imposed on many organizations is the need to make use of significant third-party products and packages as an integral part of any application. The last few years has seen a swift rise in the number of

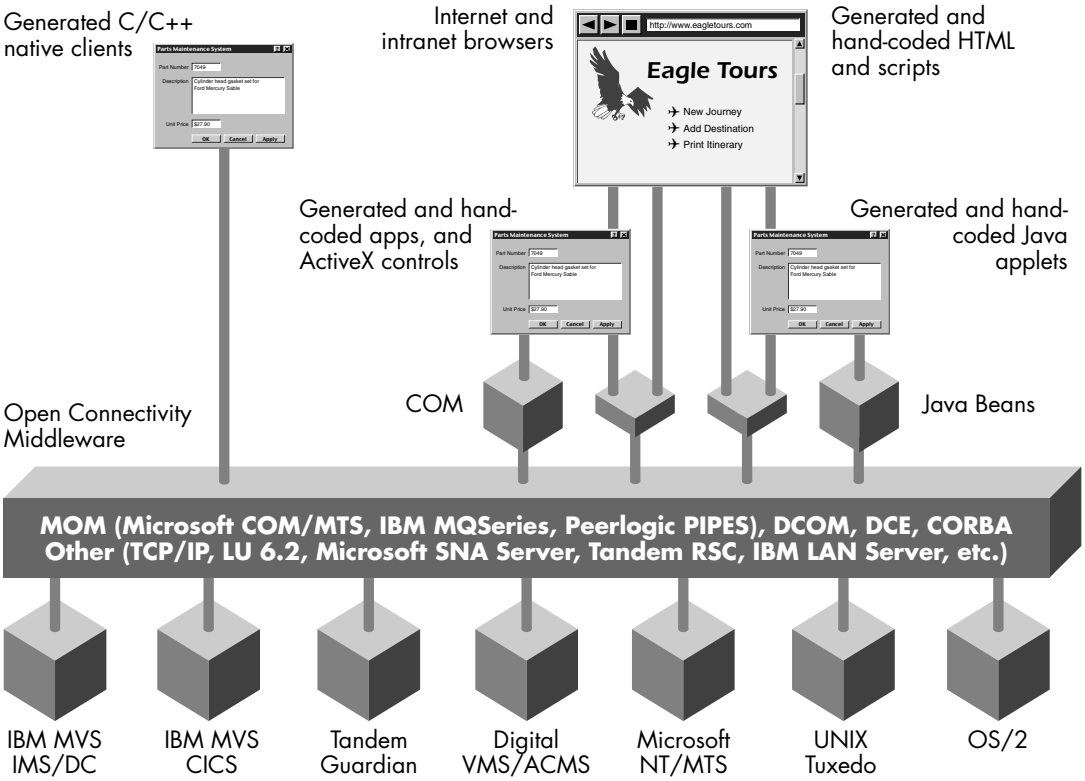

FIGURE 1.1. An example set of target deployment technologies for an application.

organizations deploying third-party developed systems, often driven by expected long-term economic benefits in reduced maintenance costs. However, the majority of organizations also maintain development and maintenance control of some aspects of their mission-critical application systems. As a result, these systems typically must interact with the third-party systems in ways unanticipated by their developers. These requirements can greatly increase the complexity of the applications portfolio as a whole. Unfortunately, lack of accessibility to details of third-party products means that the resulting requirements for integration are all too often a problem to define, difficult to specify, and very time-consuming to implement and maintain.

Adapting to Change

Change is endemic to any large system. Change can bring risks to those unprepared, or open up new avenues of business to those best able to take advantage

of the opportunities it brings. Whether an organization manages change in an effective way is a major discriminator among organizations. Those organizations able to anticipate change, predict its likely impact on the way they do business, and implement effective measures to take advantage of it, are likely to be those that flourish. The other organizations will be left to deal with the resultant chaos that ensues when the effects of change leave their well-intentioned plans and approaches rendered inefficient and impractical.

While change is inevitable, recent years have seen increased levels of change driven by a variety of forces. We distinguish three forces of change here: business, technology, and organizational change.

Business Change

As organizations compete in a market economy, those able to provide the best services at the most competitive price are likely to succeed. This requires that an organization understand the business context in which it operates, and enhances that business with the computer systems it uses to support it. As changes occur in the business environment, the computer systems themselves must change to continue to provide appropriate functionality.

Recently, a number of organizations have faced unprecedented changes to their business, driven by at least three significant trends.

First, a number of industries have recently faced changes in government policies and practices resulting in major changes to the business practices in those industries. For example, in both the United States and the European Union the banking, insurance, telecommunications, and electricity supply industries have all dealt with government deregulation at a state or federal level. This deregulation has had a major impact on business practices by introducing new forms of competition, breaking down traditional supply chain monopolies, and providing consumers with greater choice in the kinds of services they can acquire and from whom.

Second, the business community has been faced with an increasing number of acquisitions, mergers, and takeovers of organizations. While each such event results in major changes to the business practices within the organizations involved, there are also frequently changes required by a large number of organizations who partner, trade, supply, or compete with those organizations. This can impact tens if not hundreds of organizations, particularly for some of the larger mergers in the banking, defense electronics, and telecommunications industries. As the merged organizations move forward, there is typically a complete reassessment of the business practices to be enacted, a redefinition of the markets served, and new products and service offerings to be developed, marketed, and supported.

Third, a number of major political and economic events will continue to cause many organizations to review and upgrade their computer systems. The most publicized global event affecting computer systems was the Year 2000 (Y2K) problem that forced many organizations to repair many of their existing systems. To deal with Y2K, many computer systems were examined and suitable changes put in place. This is only one of many events, however, that require computer systems to be upgraded. Events such as the European Monetary Union (EMU), the creation of the North American Free Trade Agreement (NAFTA), and the expansion of the members of the North Atlantic Treaty Organization (NATO) are being faced by specific industries, or in particular regions of the world. Each such event may require significant changes to be made to mission-critical systems in response to changes in taxation, tariffs, currencies, laws of competition, and so on.

Technology Change

Some of the most far-reaching changes being faced by organizations are as a result of significant advances in technology. The rapid pace of change of computer technology is something that many people are aware of, yet for which most are largely unprepared.

Typically, an organization embarking on a large-scale software development or upgrade effort has to make a number of choices concerning how they intend to deal with the technological changes that will undoubtedly occur as the project proceeds. Almost inevitably, technology choices made early in the project life cycle will be made obsolete by technology advances that take place before the results of the project can be deployed. This is particularly evident with the latest wave of Internet-based technologies such as Java, web-browsers, and application servers. In some situations, particularly government-funded projects, one can argue that the typical life of advanced technologies is shorter than the procurement process that is needed to purchase those technologies!

This rapid pace of change is not limited to the computer industry. Many application domains being supported by computer-based systems are themselves undergoing significant technological change. Particular examples of such domains include the telecommunications, manufacturing, and entertainment industries. This leads to a curious tension between the domains in which the software is used, and the software industry itself. On the one hand, as these industries evolve they are forcing major changes to the software systems that form an essential part of their infrastructure. On the other hand, the ability of these industries to make changes is severely constrained by the difficulties of updating the software-intensive systems required to manage and support their

day-to-day activities. This dichotomy is at the heart of many of the battles currently taking place in the boardrooms of large corporations across the world.

Organizational Change

Many organizations are facing unprecedented change as they adopt new work practices and management styles encouraged and supported by the changes in business practices in those domains. Many examples of this exist—from the rise of home offices and remote working, to the relocation of call centers to reduce costs and enable around the clock operation. The dynamic, flexible way in which organizations are structured is the cause of significant kinds of change. Such change can indirectly have a major impact on the software-intensive systems being developed, and the techniques that are used for their development, deployment, and maintenance. Three particular sources of organizational change can be identified.

First, modern management tends to emphasize a much more flexible approach to structuring an organization. Many of these approaches introduce sources of change within the organization to allow appropriate teams to form, complete a task, and disperse. Typical of these approaches are techniques that emphasize matrix management, project-based integrated product teams, and "just-in-time" delivery of skills, resources, products, and services. While the dynamic nature of the organizational structure is considered an advantage in allowing projects to quickly engage in a task, it also represents a major challenge to the support technology used by that organization.

Second, the workforce employed over the past few years has needed to be much more mobile and willing to change organizations than ever before. Certainly, the days when someone was employed by one organization throughout his or her working life have all but disappeared. This phenomenon is fueled in part by what Alan Greenspan calls "creative destruction"—many organizations, in the rush to reinvent themselves or seize new opportunities, are in a constant state of rebuilding their staff and skills base. Consequently, it is not unusual for organizations to see significant turnover of staff in a given year. As a result, much of the continuity and knowledge gained by the organization may be lost whenever an employee leaves. This staff turnover is a significant source of change and becomes a major source of instability without careful management.

Third, to improve flexibility in work practices there is a renewed interest across the software industry in greater use of outside contractors and third-party service providers. The market for these services has grown in response to the business and organizational changes previously highlighted. This has

led to a host of new companies offering specialized software solutions to vertical industries, focusing on specific technologies, or providing skills in particular development techniques. An example is the set of organizations acting as systems integrators (SIs) to create custom solutions based on commercially available systems and packaged software. This is one of the main ways that organizations are looking to overcome the problems of hiring and retaining skilled staff, and the difficulties of honing those skills over time.

1.4 BUSINESS DRIVERS AND IT STRATEGY

The previous discussion highlights the challenges being faced today by the IT department of any organization responsible for the development, deployment, maintenance, and evolution of software-intensive systems. Their ability to manage complexity and rapidly adapt to change is a major determinant in whether such organizations are able to compete and succeed in the industries in which they operate.

However, it is also useful to consider the challenges as they are perceived by the various business units within an organization. In particular, consider two separate aspects of their perception.

The first is the business context in which they operate and the business objectives that drive their information technology needs. In a Software Productivity Group (SPG) Analyst Services' recent annual market survey of the Fortune 1000 IT community, respondents collectively identified their companies' top five business objectives. These results are summarized in Figure 1.2.

FIGURE 1.2. The top five IT business drivers (Source: SPG Analyst Services).

Interestingly, the highest priority objectives based on this survey are to improve the quality of the services offered in support of customers' needs, and to do so in as efficient and effective manner as possible. These results are interesting because they indicate that quality and fitness for purpose are by far more important drivers than speed of execution per se. This has important implications for the software-intensive systems supporting these businesses. Based on the discussion earlier, the key considerations in achieving these goals are the ability to:

- understand the business domain in which they operate;
- produce systems which accurately model some part of that domain;
- maintain the integrity of the system as the business domain evolves.

The second aspect of the organizations' perception is how they have translated the business objectives into specific, actionable initiatives that will have an impact on the software-intensive systems used to run their businesses. The same SPG Analyst Services' annual market survey also asked the Fortune 1000 IT community to identify their top five IT initiatives and mandates. The results from those who responded are summarized in Figure 1.3.

The results summarized in Figure 1.3 reveal that exploiting intranet and Internet technologies is the highest priority IT initiative in these organizations. Improving the quality and speed of delivered applications are also considered important. However, this is overshadowed by the need to use new technologies to reduce costs and drive new business opportunities.

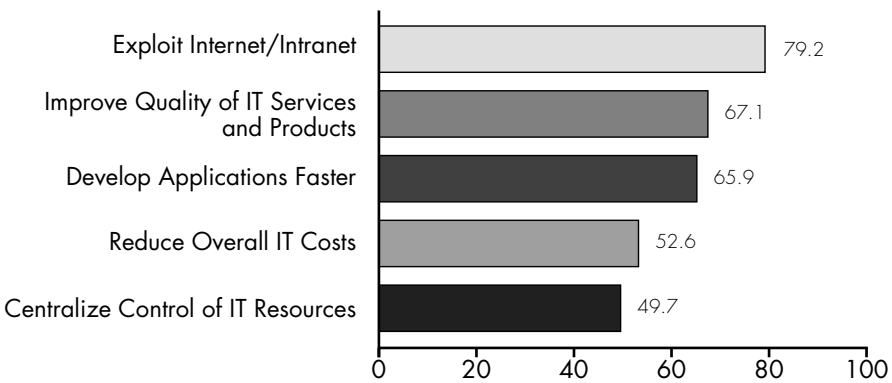

FIGURE 1.3. The top five IT initiatives and mandates (Source: SPG Analyst Services).

This survey indicates that the need to adapt to technology changes in support of improved customer satisfaction is the preeminent IT strategy today. Use of Internet-based technology is typically expected to improve the availability of desktop services to customers, increase access to corporate information sources, and reduce management effort for delivered services through thin clients and "just in time" delivery of services from the server to the desktop.

Currently, this emphasis is most often seen in two main areas. The first is in analysis, mining, and wrapping of "back-office" data and services. These are often tied up in enterprise resource planning (ERP) systems, or held in an assortment of database files. These data and services represent key corporate assets that must be made available for use with new, web-based systems. The second is in the renewed attention being paid to "front-office" customer-focused solutions. A new generation of call center management and customer relationship management (CRM) systems are being installed to offer improved response and greater personalization of interactions between an organization and its customers.

The greatest challenge facing many organizations, however, is the integration of the results of these two initiatives. This brings the whole supply-chain together, and offers the possibility of major improvements in efficiency and customer service.

1.5 SUMMARY

From its earliest days the software industry has been characterized by rapid and continual change in the domains in which software is applied, and in the underlying technologies they target and use. Coping with these changes has been a struggle for every software engineer and every software-intensive system that has been built. However, in the past few years the speed and depth of these changes has begun to increase. These changes are completely reshaping both the software industry and many of the business communities it serves. Many people believe we have seen the start of a new era of computing—the Internet age.

In this new era, software is a dominant factor in many of the current business initiatives. Software is essential for driving new Internet appliances, managing networks of computers, and supporting the businesses on which we all depend. In this capacity the software industry plays an interesting dual role. Software-led initiatives are a major catalyst for change in many domains, and provide many opportunities for new kinds of commerce or information shar-

ing. Yet at the same time the major inhibitor faced by many organizations as they try to embrace this change is the speed at which the IT infrastructure can evolve. In the face of these changes their systems are often found to be inflexible, brittle, and poorly documented.

The challenge to the current generation of software engineers is to build systems that enable and support the kinds of business initiatives typical of the Internet age. This requires approaches that:

- Make the most of previous investments in software or hardware infrastructure;
- Exploit new technologies to their full potential;
- Drive new business initiatives to provide a competitive advantage in an increasingly crowded marketplace of solutions.

Meeting these needs is the main challenge faced when provisioning enterprise-scale solutions in the Internet age.