and/or business partners focussed access to different sources of information and services through an intranet or extranet. Marketplace portals are divided into horizontal and vertical market places. Horizontal marketplaces operate on the business-to-consumer market offering consumer goods directly to the general public, and in business-to-business, selling their products to companies from other sectors. Vertical marketplaces consist of companies from a single sector, e.g. suppliers on one side and manufacturing companies on the other. Community portals are directed at specific target groups, e.g. young people, and try to create customer loyalty through user interaction or to provide individual offers through appropriate user management (one-to-one marketing).

The increasingly important category of *ubiquitous Web applications* provides customized services anytime anywhere and for any device, thus facilitating ubiquitous access. An example of this would be displaying the menu of the day on the mobile devices of all users entering a restaurant between 11 am and 2 pm. For this type of system it is important to take into account the limitations of mobile devices (bandwidth, screen size, memory, immaturity of software, etc.) and the context in which the Web application is currently being used. Based on this dynamic adjustments according to the users' situation (Kappel et al. 2002) can be made. Currently existing Web applications of this type usually offer a very limited form of ubiquity only supporting one aspect – either personalization or location-aware services or multi-platform delivery (Kappel et al. 2003).

Current developments, however, especially the increasing convergence of the TIMES industry (Telecommunications, Information technology, Multimedia, Education and Entertainment, Security), will lead to a situation in the near future where ubiquitous applications will dominate the market. One of these developments is the *Semantic Web*. The goal of the Semantic Web is to present information on the Web not merely for humans, but also in a machine-readable form (Berners-Lee et al. 2001). This would facilitate knowledge management on the Web, in particular the linking and reuse of knowledge (content syndication), as well as locating new relevant knowledge, e.g. by means of recommender systems. Through increased interoperation on the semantic level and the possibility of automating tasks (via software agents), we believe the Web will become even more ubiquitous and therefore relevant for everyday life.

1.3 Characteristics of Web Applications

Web applications differ from traditional, non-Web-based applications in a variety of features worth looking into. These are characteristics that traditional applications lack completely (e.g. non-linear navigation) on the one hand and characteristics that are of particular importance in Web applications on the other hand (e.g. frequency of updates) (Balasubramaniam et al. 2002, McDonald and Welland 2001b, Whitehead 2002). Whether a certain characteristic is present and to what degree depends partly on the type of Web application: the development of transactional Web applications such as e-commerce systems requires greater focus on the content being up to date and consistent as compared with pure information provision systems – e.g. virtual exhibitions. These characteristics are the reason why many concepts, methods, techniques, and tools of traditional Software Engineering have to be adapted to the needs of Web Engineering or may even be totally inadequate. Fig. 1-2 gives an overview of these characteristics and arranges

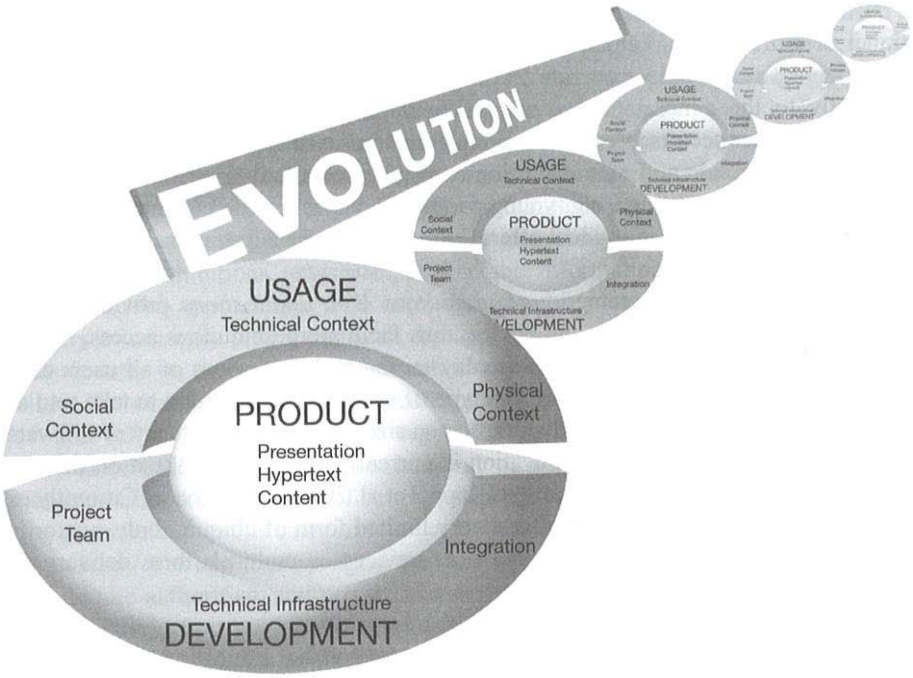

Figure 1-2 Dimensions according to ISO/IEC 9126-1 for the categorization of characteristics of Web applications.

them along the three dimensions: "product", "usage", and "development" with their "evolution" as an encompassing dimension.

These dimensions are based on the ISO/IEC 9126-1 standard for the evaluation of software quality characteristics (`http://www.iso.org/`). By assigning the different characteristics of Web applications to these dimensions we can also see their influence on the quality of applications and thus take the characteristics as a starting point for the definition of Web Engineering requirements (cf. Section 1.4). In addition to product-related, usage-related, and development-related characteristics, we have evolution as a fourth dimension governing the other three dimensions. Products must be adaptable, new contextual information should be considered during use, and development faces continually changing conditions, to name but a few examples. In the following, we will describe the individual characteristics according to these dimensions. References are made to those chapters of the book expanding on the characteristic in question. An overview of the influences of these characteristics and their occurrence in the chapters of this book is given in Table 1-1.

1.3.1 Product-related Characteristics

Product-related characteristics constitute the major building blocks of a Web application, consisting of content, the hypertextual structure (navigational structure), and presentation (the

Table 1-1 Web Engineering requirements

Characteristics of Web Applications		Characteristic \ Chapter	Web Engineering 2: Requirements Engineering	3: Modeling	4: Architectures	5: Technology-Aware Design	6: Technologies	7: Testing	8: Operation and Maintenance	9: Project Management	10: Development Process	11: Usability	12: Performance	13: Security	14: Semantic Web
Product	content	document character / multimediality		x	x		x					x			
		quality aspects	x	x				x	x			x			
	hypertext	non-linearity		x		x		x				x	x		
		disorientation / cognitive overload		x								x	x		
	presentation	aesthetics	x	x		x				x		x			
		self explication	x	x						x		x			
Usage	social context	spontaneity		x	x					x			x		
		multiculturalism	x	x				x				x	x		
	technical context	quality of service	x	x				x				x	x		
		multi-platform delivery	x	x	x	x	x	x				x	x		
	natural context	globality		x	x			x						x	
		availability		x				x			x				
Development	team	multidisciplinarity	x					x		x					x
		youthfulness						x		x					
		community development				x	x							x	x
	infrastructure	inhomogeneity		x	x	x	x	x				x			x
		immaturity			x	x	x		x	x					
	process	flexibility	x					x		x	x				
		parallelism								x	x				
	integration	internal integration	x		x	x					x	x			
		external integration	x		x	x					x	x			x
Evolution		continuous change	x		x			x	x		x	x			
		competitive pressure			x	x		x	x		x				
		short lifetime			x			x	x	x	x	x	x		

user interface). Following the object-oriented paradigm, each of these parts has not only a structural or static aspect, but also a behavioral or dynamic aspect.

Content

Generating content, making it available, integrating, and updating it is equally important as the development and provision of the actual software of a Web application. Web applications are used expressly because of the content they offer – true to the motto "Content is King". Web application developers must therefore not only act as programmers but also as authors. Important aspects are the varying degree of structure of the content and the quality demands users make on the content.

- *Document-centric character and multimediality*: Depending on the structuring, content is provided as tables, text, graphics, animations, audio, or video. "Document character" in Web applications refers to the fact that content is provided, i.e. documents are generated that present information in an appropriate way for certain user groups (e.g., tourist information on a holiday region). This implies amongst others special requirements on usability (cf. Chapter 11, Usability). Content is in part also generated and updated dynamically; e.g. the number of available rooms in a tourism information system. Furthermore, the Web serves as an infrastructure for the transmission of multimedia content, e.g. in video conferences or Real Audio applications.
- *Quality demands*: Depending on the application area, the content of a Web application is not only subject to differing update frequencies, but also to different quality metrics regarding its being up to date, exact, consistent and reliable. This requires not only the consideration of these quality demands in the requirements definition (see Chapter 2, Requirements Engineering), but also the evaluation of compliance with these principles (see Chapter 7, Testing).

  News sites, for instance, have a very high frequency of updates and face very high user demands regarding topicality. The Web as a medium in its own right, alongside television, radio, and print media, offers great potential for addressing these demands better than traditional media, e.g. through personalization. On the other hand, there is a line of argumentation saying that "smart", i.e., location aware, personalized applications also require for new genres to be developed: the reason is that these new content-driven applications such as podcasting or mobile contents are such a different medium that one cannot simply adapt existing content but that rather new genres have to be developed in order to provide high quality of user perception (see also "Content Engineering" Reich and Güntner 2005).

  Particularly high quality is required for price and availability information in online-shopping systems, as they form the basis of the business transaction (cf. e.g. Pröll and Retschitzegger 2000). Incorrect prices can lead to a cancellation of the sale, out-of-date information on availability can result in products on stock not being sold or in delivery problems because products listed as available are not on stock after all.
  Regardless of where a Web application is used, content quality is a critical factor for its acceptance. The great challenge is being able to guarantee the quality of the data despite the large volume and high frequency of updates.

Hypertext

Amongst the specific characteristics of Web applications is the non-linear nature of hypertextual documents (Conklin 1987). The hypertext paradigm as a basis for the structuring and presentation of information was first mentioned by Vannevar Bush (Bush 1945). There are many different hypertext models (McCarty 2003), and the Web itself defines a very simple model of its own. Basic elements of hypertext models are *nodes, links* and *anchors*. A node is a self-contained uniquely identifiable information unit. On the Web this might be an HTML document which can be reached via a URL (*Uniform Resource Locator*). A link is the path from one node to another. On the Web, these paths are always unidirectional and their meaning is not clearly defined. Possible meanings include "next node according to recommended reading order" or "diagram for mathematical formula". An anchor is the area within the content of a node that is the source or destination of a link, e.g. a sequence of words in a text or a graphical object in a drawing. On the Web, anchors are only possible in HTML documents.

The essential feature of the hypertext paradigm is the *non-linearity* of content production by the authors and of content reception by the users together with the potential problems of *disorientation* and *cognitive overload.*

- *Non-linearity*: Hypertexts imply stereotypes of relatively systematic reading, and in this, Web applications differ fundamentally from traditional software applications. We can distinguish among others between *browsing*, e.g. in online shopping applications, *queries*, e.g. in virtual exhibitions, and *guided tours*, e.g. in e-learning applications. This individual style of reading, adaptable to user needs and behavior, is ideally suited to the human learning ability. Users may move freely through the information space, depending on their interests and previous knowledge. Anchors (and, consequently, also links) are not only predefined statically by the authors, but are also generated dynamically (computed links) in a predefined reaction to user behavior patterns. Creating hypertext is always a challenge for the authors, as they seek to avoid disorientation and cognitive overload for the users (see Chapter 3, Modeling, and Chapter 11, Usability).
- *Disorientation and cognitive overload*: It is particularly important in Web application development to cope with these two fundamental problems of the hypertext paradigm. Disorientation is the tendency to lose one's bearings in a non-linear document. Cognitive overload is caused by the additional concentration required to keep in mind several paths or tasks simultaneously. Sitemaps, key word searches, retracing of "paths" (*history mode*) and display of access time and time spent on the site help users to keep their orientation within the application. Meaningful linking and intelligent link naming reduce cognitive overload (Conklin 1987). Additionally, design patterns in modeling the hypertext aspect may also help counteract this problem (Akanda and German 2005, German and Cowan 2000, Lyardet and Rossi 2001, Panagis et al. 2005) (see Chapter 3, Modeling, and Chapter 11, Usability).

Presentation

Two special features of Web applications at the presentation level, i.e. the user interface, are aesthetics and self-explanation.

- *Aesthetics*: In contrast to traditional applications, the aesthetics of the presentation level of a Web application, the "look and feel" of the user interface, is a central factor not least because of the high competitive pressure on the Web. The visual presentation of Web pages is subject to fashion trends and often determines success or failure, in particular for e-commerce applications (Pressman 2005).
- *Self-explanation*: Besides aesthetics, it is essential that Web applications are self-explanatory, i.e. it should be possible to use a Web application without documentation. The navigation system or interaction behavior must be consistent within the whole application, so that users can quickly become familiar with the usage of the Web application (see Chapter 11, Usability).

1.3.2 Usage-related Characteristics

Compared with traditional applications, the usage of Web applications is extremely heterogeneous. Users vary in numbers and cultural background, devices have differing hardware and software characteristics, and the time and location from where the application is accessed cannot be predicted (Kappel et al. 2000). Additionally, developers not only have no possibility of knowing the potential diversity of these so-called *contextual factors* in advance, they also cannot influence them in any way because of their autonomous nature. There is hardly any way of predicting for example the usage frequency for a given Web application (see Chapter 2, Requirements Engineering, and Chapter 12, Performance).

The usage of Web applications is therefore characterized by the necessity to continuously adapt to specific usage situations, so-called *contexts*. Adjustment to these contexts can be equally necessary for all parts of the Web application, i.e. content, hypertext, and presentation (see Chapter 3, Modeling). Because of the fundamental significance of adjustment to contexts, usage-related characteristics are divided into three groups: *social context*, *technical context*, and *natural context* (Kappel et al. 2000, Koch and Wirsing 2001, Kappel et al. 2003).

Social Context: Users

The social context refers to user-specific aspects; spontaneity and multiculturality in particular create a high degree of heterogeneity.

- *Spontaneity*: Users can visit a Web application whenever they want and leave it again – possibly for a competitor's site. The Web user cannot be expected to be loyal to any content provider. The Web is a medium that entails no obligation (Holck and Clemmensen 2002). Since it is easy to find competing applications with the help of search engines users will only use a Web application if it appears to bring them immediate advantage.

  Spontaneity in use also means that the number of users cannot be reliably predicted as for traditional applications. Scalability, therefore, is extremely important (Hendrickson and Fowler 2002) (see Chapter 4, Architecture, and Chapter 12, Performance).
- *Multiculturality*: Web applications are developed for different user groups. If the group in question is a known user group, as would be the case with an intranet or extranet,

sentation level of
l factor not least
entation of Web
, in particular for

ːations are self-
ːumentation. The
/hole application,
application (see

ːxtremely hetero-
iffering hardware
ːation is accessed
ave no possibility
dvance, they also
ere is hardly any
ɔn (see Chapter 2,

y to continuously
: contexts can be
, and presentation
tment to contexts,
ınical context, and
)3).

rality in particular

/ant and leave it
pected to be loyal
gation (Holck and
ı the help of search
g them immediate

ːliably predicted as
t (Hendrickson and
ːe).
ːoups. If the group
ıtranet or extranet,

this is largely comparable to traditional applications. When developing a Web application for an anonymous group of users, however, there will be large and hardly foreseeable heterogeneities in terms of abilities (e.g. disabilities), knowledge (e.g. application expertise), and preferences (e.g. interests) (Kobsa 2001). In order to allow appropriate customization, assumptions about the user contexts must be made at the development stage of a Web application. These will be taken into consideration when adapting the components of the application. Regular customers might be given special discounts (adaptation of content), new customers might receive a guided tour through the Web application (adaptation of hypertext), and users with visual impairments might be aided by appropriate font sizes (adaptation of presentation). Personalization often requires users to set their preferences (e.g. preferred payment method on `http://www.amazon.com`).
The large variety of possible user groups also makes it hard to define a representative sample for a requirements analysis (see Chapter 2, Requirements Engineering).

Technical Context: Network and Devices

The technical context comprises properties relating to the network connection concerning *quality of service*, and the hardware and software of the devices used to access the Web application, for *multi-platform delivery*.

- *Quality of service*: Technically, Web applications are based on the client/server principle. The characteristics of the transmission medium, such as bandwidth, reliability, and varying stability of the connection are independent factors that must be considered when developing a Web application to guarantee appropriate quality of service (Badrinath et al. 2000, Pressman 2005). For example, the parameter "maximum bandwidth" can be adjusted to optimize the amount of data transferred, so that multimedia content, e.g. videos, will be transferred with lower resolution in case of lower bandwidth. While for traditional applications the specifications of the network are usually known beforehand, Web application developers need to make assumptions about these properties (see Chapter 7, Testing, and Chapter 12, Performance). With the trend towards mobile Web applications, this is of increasing importance, as convergent networks require even more adaptation on the application level (Venkatakrishnan and Murugesan 2005).
- *Multi-platform delivery*: Web applications usually offer services not only to a specific type of device, but rather any, increasingly mobile, devices with very different specifications (e.g. monitor size, memory capacity, installed software) (Eisenstein et al. 2001). The large number of different browser versions is also a challenge, as they have different functionalities and restrictions (and also often do not implement the specifications as expected). This poses difficulties in creating a consistent user interface and in testing Web applications (see Chapter 7, Testing).

  Additionally, users can configure browsers autonomously. Presentation (e.g. hide images), access rights (e.g. for Java applets), and range of functions (e.g. cookies and caching) can all be configured individually, thus having an influence on performance, transaction functionality, and possibilities of interaction, to name but a few (see Chapter 4, Architecture, Chapter 5, Technology-aware Design, and Chapter 6, Technologies).

Based on assumptions of typical classes of devices, Web application developers can adapt content to PDAs (*personal digital assistants*) by not transmitting images or videos (*web clipping*) and instead providing links or descriptive text. At the hypertext level, printer versions of hypertext documents can be provided. Finally, in order to account for different versions of JavaScript in different browsers, platform-independent libraries can be used in the development process (see e.g. `http://www.domapi.com`).

Natural Context: Location and Time

The natural context includes aspects of the location and time of access. Globality and availability create a high degree of heterogeneity.

- *Globality*: The location from which a Web application is accessed, e.g. the geographical position, is important for the internationalization of Web applications regarding regional, cultural and linguistic differences. Additionally, the (physical) location can be used in conjunction with location models to define a logical position such as place of residence or workplace in order to provide location-aware services. Location-awareness imposes further difficulties for the testing of Web applications as it is often hard to simulate changing locations and/or test all possible locations (see Chapter 7, Testing). Global availability also increases the demands on security of Web applications to prevent users from accessing – deliberately or by accident – private or confidential areas (see Chapter 13, Security).
- *Availability*: The "instant delivery mechanism" inherent in the very nature of the Web makes the application immediately available. The Web application becomes instantly usable, which means that the quality of the developed product must be secured. Permanent availability 24/7 also increases the demands on the stability of Web applications (see e.g. Chapter 7, Testing). In addition, time-aware services are made possible through consideration of the time aspect (e.g. timetable information depending on the time of day and day of the week).

1.3.3 Development-related Characteristics

The development of Web applications is characterized by the necessary resources, such as the *development team* and the *technical infrastructure*, the *development process* itself, and the necessary *integration* of already existing solutions.

The Development Team

The development of Web applications is strongly influenced by the fact that development teams are *multidisciplinary* and generally rather young. These factors and the methods of the so-called *community development* contribute to a completely new way of organizing collaboration of different groups of developers. The different points of view and emphases must be brought together through appropriate project management and an adapted development process (see Chapter 9, Web Project Management, and Chapter 10, Development Process).

evelopers can
ıges or videos
:t level, printer
nt for different
can be used in

ınd availability

ıe geographical
ırding regional,
can be used in
ce of residence
ıreness imposes
ard to simulate
'esting). Global
ions to prevent
ntial areas (see

ture of the Web
:comes instantly
ured. Permanent
ıpplications (see
ɔossible through
ı the time of day

:sources, such as
ess itself, and the

evelopment teams
ds of the so-called
g collaboration of
; must be brought
ment process (see
.

- *Multidisciplinarity*: Web applications can be characterized as a combination of print publishing and software development, marketing and computing, and art and technology (Powell et al. 1998). Therefore, the development of Web applications should be perceived as a *multidisciplinary approach* requiring knowledge and expertise from different areas. In addition to IT experts responsible for the technical implementation of the system, hypertext experts and designers should be employed to design hypertext and presentation, while domain experts should be responsible for the content. There is therefore a larger variety of competence and knowledge in the development team than in traditional software development (see Chapter 5, Technology-aware Design).

  Which discipline will dominate depends on the type of Web application. While e-commerce applications are based more on traditional database and programming expertise, developing a virtual exhibition would put more emphasis on domain and design expertise.
- *Young average age*: Web application developers are on average significantly younger – and thus less experienced – than traditional software developers. They usually live up to the stereotype of the "technology freak" who does not care too much about old conventions and is very interested in new tools and technologies (McDonald and Welland 2001b).
- *Community development*: The development of open source software freely available on the Web and its integration in "real" applications is a very recent phenomenon. Developers use this software for their own developments, which they in turn make available for the open source community. The conscious inclusion of external developers or groups of developers with their unwritten laws of cooperation is an important feature of this new form of community development (see Chapter 6, Technologies).

Technical Infrastructure

The *inhomogeneity* and *immaturity* of the used components are important characteristics of the technical infrastructure of Web applications (see Chapter 4, Architecture, and Chapter 5, Technology-aware Design).

- *Inhomogeneity*: The development of Web applications depends on two external components: server and browser. While the Web server can usually be configured and operated as desired by the application programmers, there is no way to influence the users' Web browsers and their individual preferences. This situation is additionally complicated by different browser versions and their inter-operation with plug-ins (see section 1.3.2, Technical Context).
- *Immaturity*: Because of the increasing time-to-market pressure, components used in Web applications are often immature, i.e. they either have bugs or lack the desired functionality. Additionally, a version update of the Web application often entails a change of the development environment. As a result, development knowledge is often lost or cannot even evolve in the first place.

Process

The development process is the framework for all development-related characteristics, and is in turn influenced by *flexibility* and *parallelism* (see Chapter 9, Web Project Management, and Chapter 10, Development Process).

- *Flexibility*: In Web application development it is impossible to adhere to a rigid, predefined project plan. It is vital to react flexibly to changing conditions.
- *Parallelism*: Due to the necessity for short development times and the fact that Web applications can often be split up into autonomous components (e.g. authentication, search function, news ticker, etc.), many Web applications are developed in parallel by various subgroups of the development team. Contrary to traditional software development these subgroups are therefore structured according to these components and not according to the expertise of the project members (e.g. GUI developers, data modelers, etc.) (McDonald and Welland 2001b).

  In addition to this *parallel development of application parts*, methodical tasks such as design, implementation and quality assurance are often carried out simultaneously for different versions. For example, quality assurance might be in process for an earlier version, while implementation has already begun for the next version and the following version is already being designed. This *parallel running of phases* poses new requirements for the planning of deployment of developers in Web projects.

Integration

A special characteristic of many Web applications is the need for *internal* and *external integration*. Integration in this context refers not only to technical aspects (see Chapter 4, Architecture, Chapter 5, Technology-aware Design, and Chapter 6, Technologies), but also to content (see Chapter 14, Semantic Web), and organizational aspects (see Chapter 10, Development Process).

- *Internal integration*: Frequently, Web applications have to be integrated with existing legacy systems when existing content, e.g. product catalogues, are to be made available through a Web application.
- *External integration*: In addition to internal integration, the integration of content and services of external Web applications is a special characteristic of Web applications. Despite strong similarities to heterogeneous database systems there are a number of particularities in integration on the Web (Lowe and Hall 1999, Sattler et al. 2002). First of all, there are a very large number of sources, frequently changing and with a high degree of autonomy concerning availability and schema evolution. Additionally, usually only few details are known about the properties of these sources, e.g. their content or functionalities. And finally, the different sources are often very heterogeneous at various levels, be it at the data level, the schema level, or the data model level.

  The integration of external services, e.g. in portal-oriented Web applications, is based on the increasingly common development form of providing and using Web services (Weerawarana et al. 2005). A Web service in this context is a reusable component with an unambiguously defined interface and functionality. The interaction of different Web services, avoiding undesired side effects, and guaranteeing quality of service are but a few of the many relevant issues in this context.

1.3.4 Evolution

As mentioned above, evolution is a characteristic that governs all three dimensions of product, usage and development (see especially Chapter 8, Operation and Maintenance). The need for evolution can be argued for with the *continuous change* of requirements and conditions, the *competitive pressure*, and the general *fast pace* of development.

- *Continuous change*: Web applications change rapidly and are therefore subject to permanent evolution due to constantly changing requirements or conditions (Scharl 2000). The rapid and never-ending change of Web technologies and standards in particular makes it necessary to continuously adapt Web applications to these. This has two reasons: users want the newest Web hype, and the used tools are also technology-driven. This constant change of requirements and conditions is a central characteristic of Web applications. Changes may concern all three dimensions of a Web application – the product itself, its usage, and, in particular, its development.
- *Competitive pressure*: The extremely high competitive pressure on the Web, the time-to-market pressure and the necessity for a Web presence (comparable to the gold rush of the late 1840s (Murugesan 2000), increase the need for ever *shorter product lifecycles* and extremely *short development cycles* and apparently leave no room for a systematic development process. Immediate Web presence is considered more important than long-term perspective (Pressman 1998).
- *Fast pace*: The extreme time pressure on Web application development is due to the rapid change on the Web and the accordingly short lifespans of Web applications or their frequency of updates. Tsichritzis sums it up very aptly in (Tsichritzis 2000): "either you are fast or irrelevant".

While for conventional software, evolution takes place in a planned series of versions, it is continuous for Web applications. This means that Web applications are in permanent maintenance. The cycle of change is often no longer than a few days or weeks (Pressman 2005). Web applications therefore require "lean" versions of traditional Software Engineering processes with special emphasis on requirements analysis and specification (Chapter 2) on the one hand and operation and maintenance (Chapter 8) on the other.

1.4 Objectives and Structure of the Book

The objectives of this book can be defined as follows:

- Provision of insight into current concepts, methods, techniques, tools, and experiences for an engineering approach to Web application development.
- Identification of similarities and differences between the development of traditional (non-Web-based) applications and the development of Web applications.
- Analysis of concepts, methods, techniques, and tools of traditional Software Engineering to see how suited they are for Web application development.
- Exploration of potential risks in Web application development.
- Outlook on future developments in Web Engineering.

The structure of this book is based on that of the *Guide to the Software Engineering Body of Knowledge* (SWEBOK, Bourque and Dupuis 2005), a compendium of the different activities of traditional Software Engineering. These activities are also applicable to Web application development, although – as this book shows – the details and their sequence and schedule have