Presentation
Paper
Bio

Return to Main Menu

P R E S E N T A T I O N

T1

Thursday, Dec 7, 2000

e-Testing: What e-Xpertise Do You Need

Tim Koomen

International Conference On
Software Testing, Analysis & Review
DEC 4-8, 2000 • COPENHAGEN, DENMARK

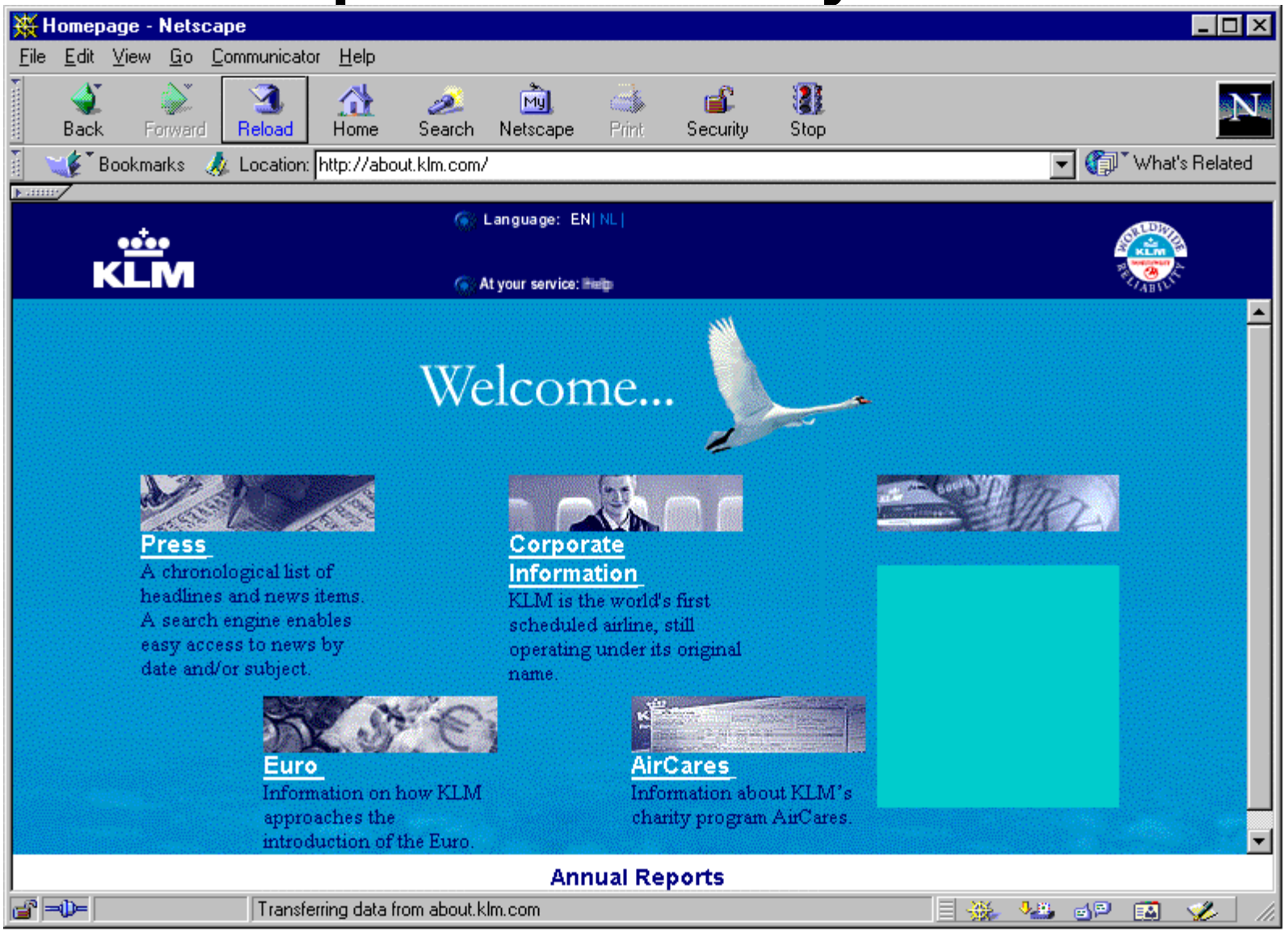
E-Testing

Tim Koomen
IQUIP Informatica B.V.
The Netherlands
t.koomen@iquip.nl

Agenda

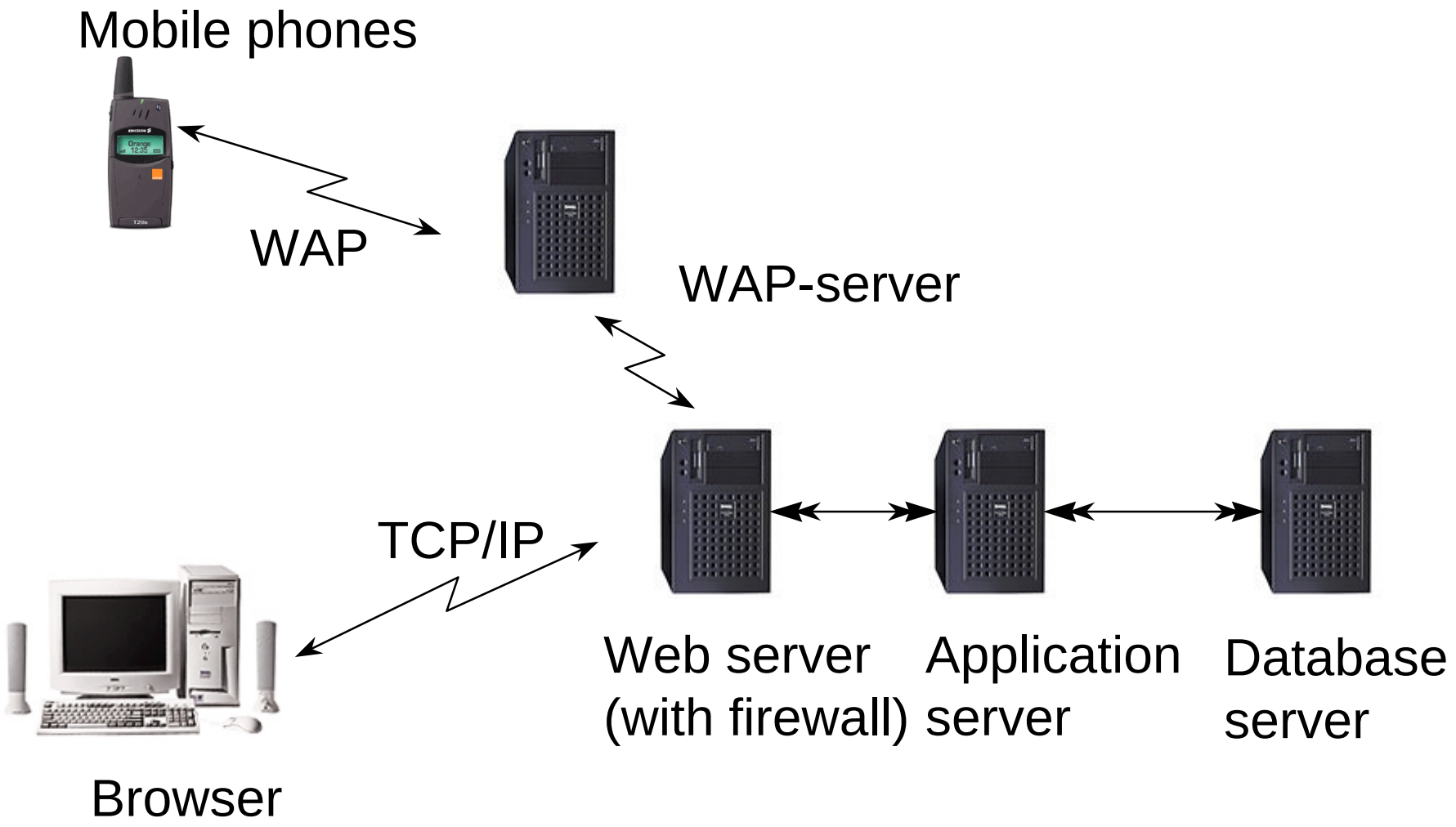
- Internet (r)evolution
- Why test e-business?
- How to test e-business?

From simple functionality ...

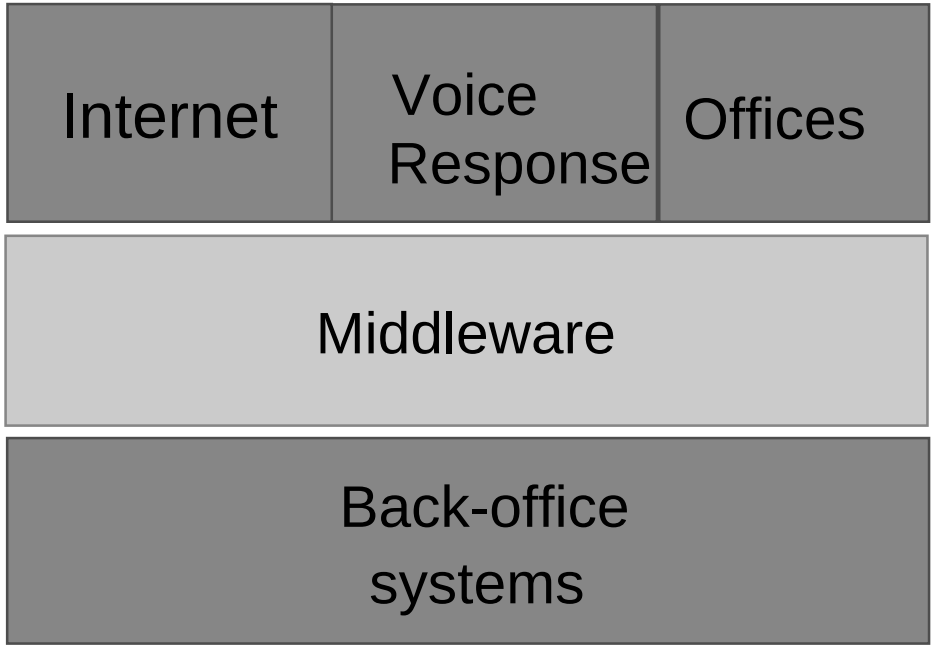


… to complex functionality

Internet (infra)structure 1



Internet (infra)structure 2



Agenda

- Internet (r)evolution
- Why test e-business?
- How to test e-business?

E-business risks

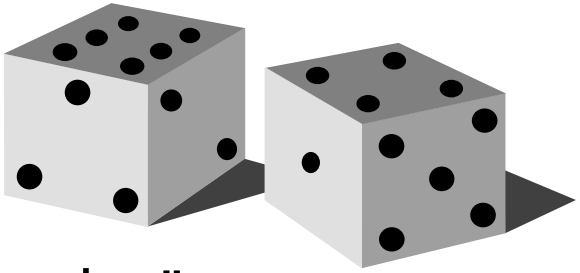
- Site is too slow or is down
- User unfriendly
- Hacker attack
- Infrastructural restrictions (browsers!)
- Insufficient organisation
- Faulty processing

Why test e-business?

- Historically: little testing
- Increasing risks
- Necessity of testing increases:
  Testing gives insight in quality of software
- Test strategy =

  testing versus risks (for every quality attribute)
- Risk = chance of failure * damage

Risk: chance of failure factors

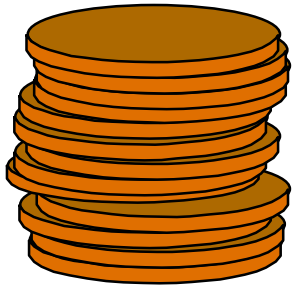


- Ever increasing complexity
  - integration with back-office
  - more functionality
- Time-to-market, “delivery due yesterday”
- Insufficient knowledge and experience:
  - developers and testers
  - customer
  - specific aspects like security and usability
- Insufficient documentation
- New working method for organisation
- Unknown users, hardware, software
- Continuous change

BUT: use of many standard or existing components!

Risk: damage factors

- Damage gets higher
  - more transactions
  - transaction is of higher value
  - missing "market share", because users
    - can’t find site
    - don’t do business
    - lose confidence and don’t return



Agenda

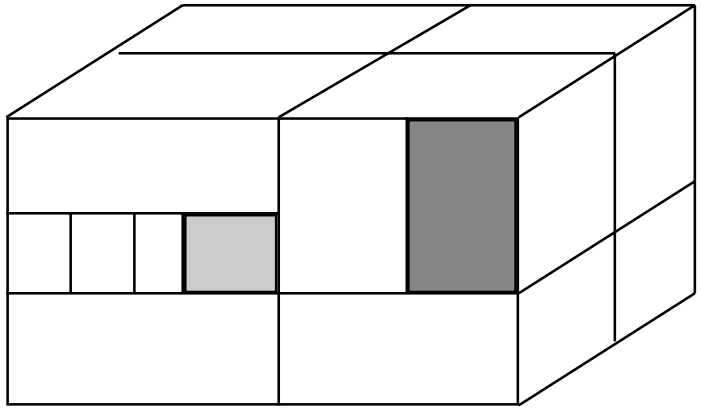
- Internet (r)evolution
- Why test e-business?
- How to test e-business?

Testing Strategy

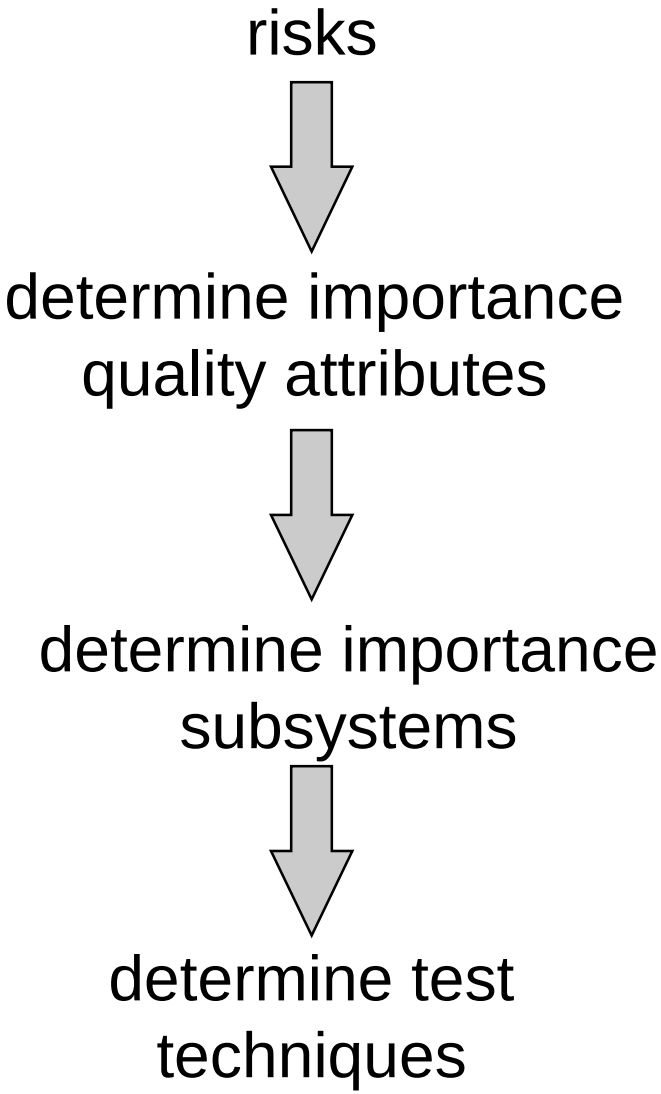
Aim: To detect the most important defects as soon as possible at the lowest price!

Dependencies:

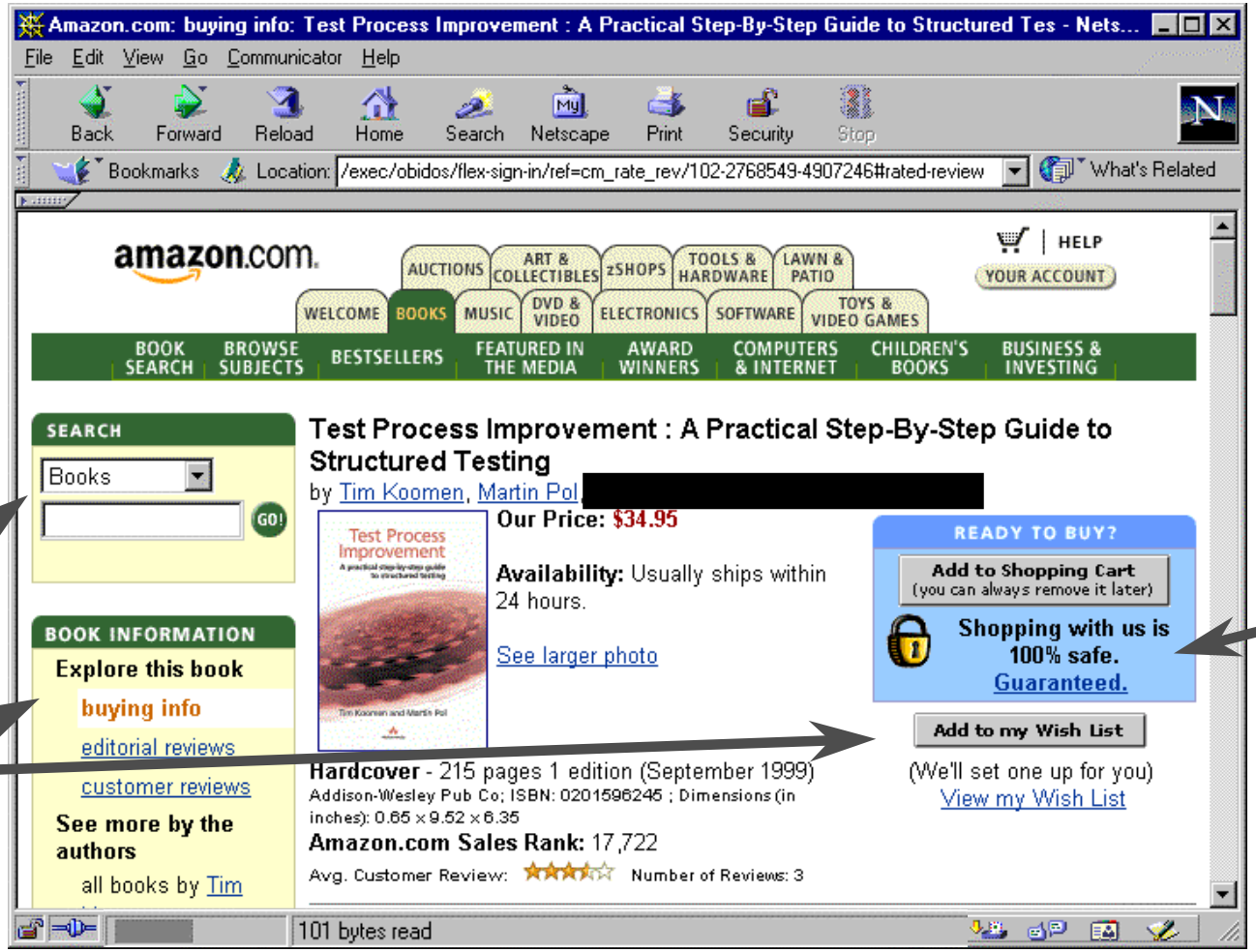
- Risks (necessity)
  - business
  - project
  - test
- Quality attributes
- Available resources



Test strategy



Example quality attributes: Amazon.com



Portability

Security

Usability
User friendliness

Time behaviour, Reliability

Quality attributes

- Which are (in general) of importance?

Dynamic

- ⇨ security (functionality)
- ⇨ accuracy (f)
- ⇨ suitability (f)
- ⇨ usability
- ⇨ reliability
- ⇨ time behaviour (efficiency)

resource utilisation (e)

Static

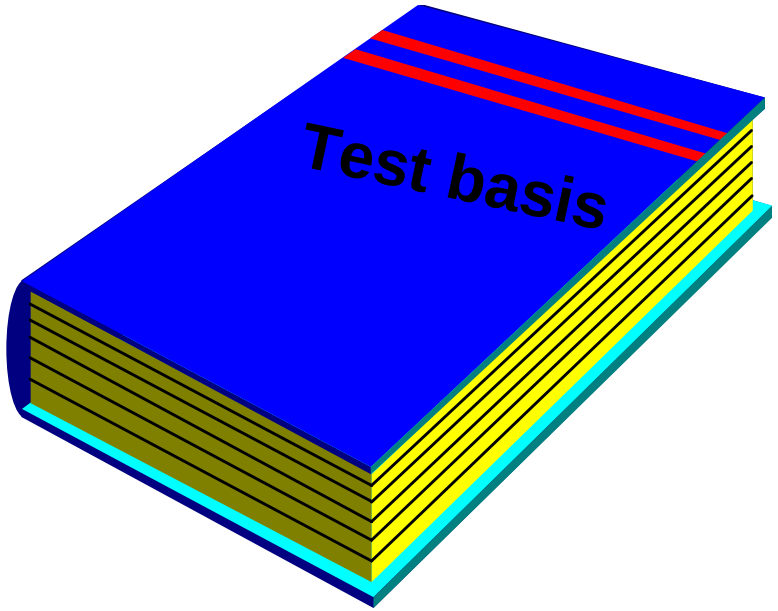
interoperability (f)

- ⇨ maintainability
- ⇨ portability
- ⇨ operability (usability)

ISO 9126

Test basis

- Not readily available
- Kinds:
  - Requirements (business scenarios, use cases)
  - FD/TD (often very “light”)
  - Norms & standards
  - External references
- Much “digging” required, interviews
- Sometimes measuring i.s.o. testing

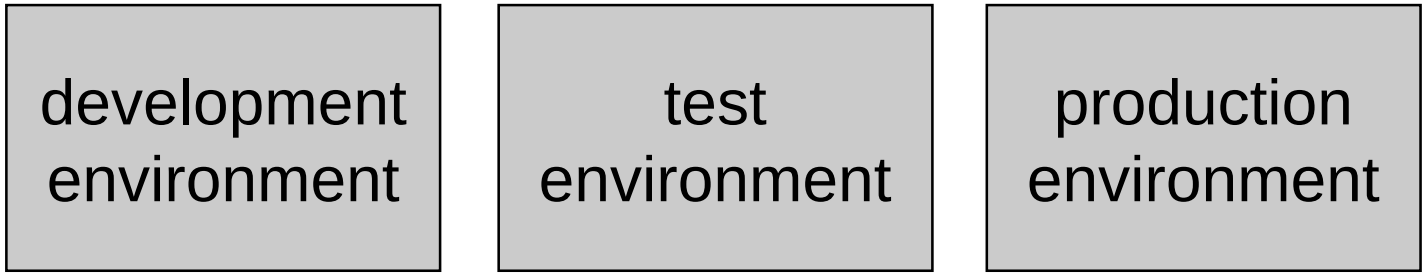


Test techniques

Many checklists

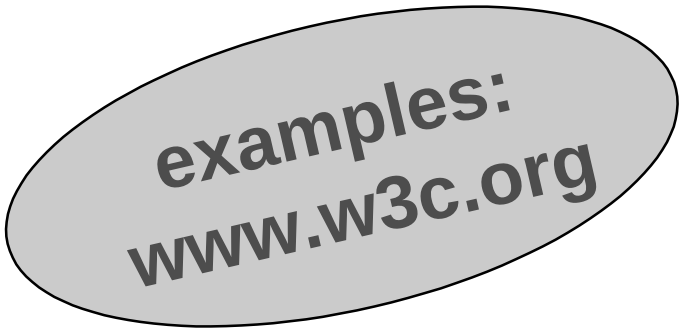
- Informal techniques: error guessing, DFT
- Performance test
- Security: audit, hacker software, penetration test
- Usability
  - lab with simulation
  - SUMI and WAMMI checklists

Test environment



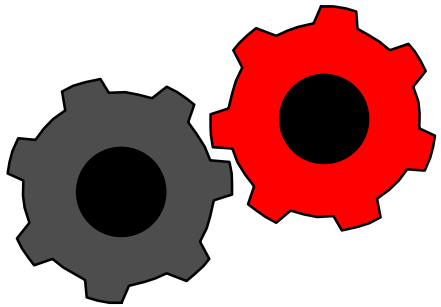
Test automation 1

- Test execution:
  - performance (also peak)
  - functionality (regression)
  - portability to browsers (versions and settings)
  - monitoring
- Scanning HTML-scripts to check:
  - syntax
  - portability (browsers)
  - size of images
  - spelling
  - external links



Test automation 2

- Test tools
  - Load & Stress
  - Internet specific C&Pb, big brands + new brands
  - challenges:
    - moving pictures with applets, and so on
    - number of changes in pages
  - HTML-check tools (some free, some cheap via the Web)



Knowledge and skills

- Testing
- Business: specs often missing!
- Internet, system and architecture: what is standard?
- Test tools: Capture&Playback + Load&Stress + HTML-tool
- Specific (for instance, security and performance)

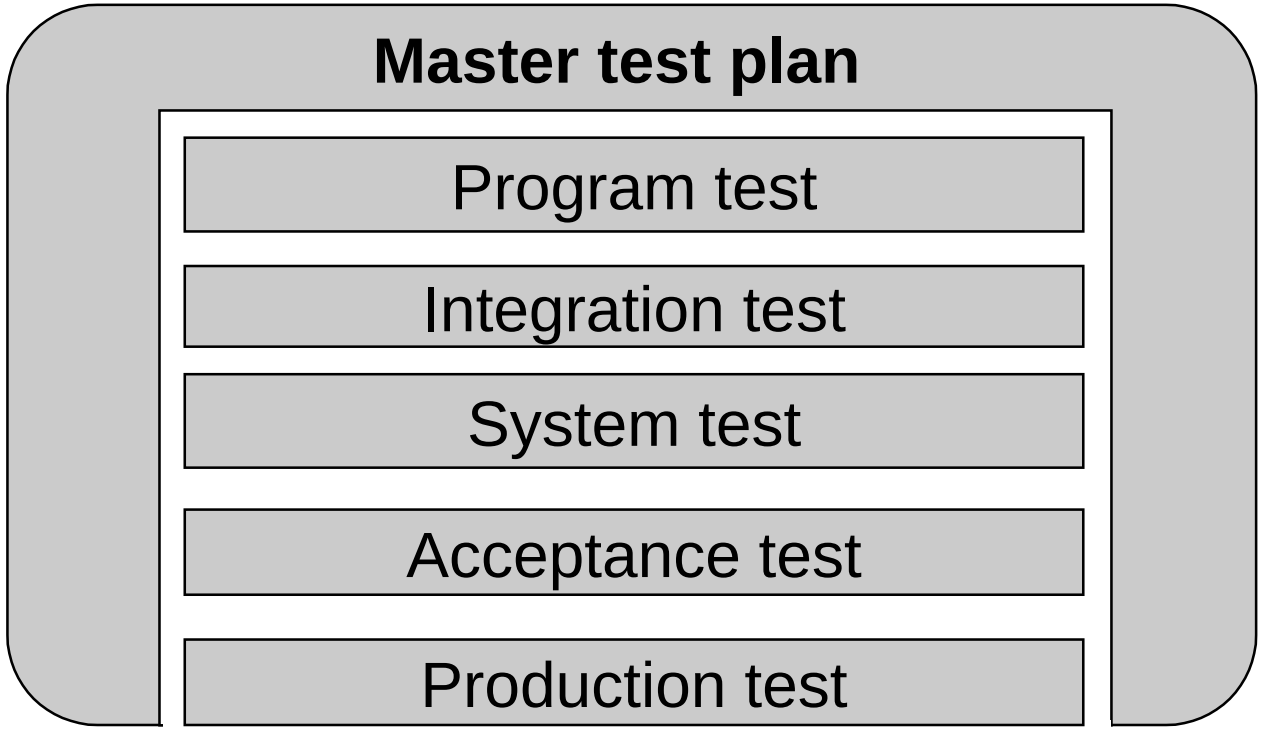
And we keep on testing … the production test

Periodically testing:

- Performance monitoring
- Integrity check
- Security monitoring
- and so on

TESTING as control activity

Test levels in master test plan



NEW: production test during exploitation phase

Test trends

- More mature development process
  - More requirements and specifications
  - More test environments
  - Configuration management
- TMap keeps helping...
  - Life-cycle model with testing strategy
  - Techniques
  - Infrastructure with tools
  - Organisation

Conclusion

- E-business testing requires its own test approach
  - Focus on testing non-functional quality attributes
  - Test automation very important
  - Black-box testing and white-box testing mix
- Specific kinds of e-xpertise necessary

THANK YOU!

E-Testing, what e-xpertise do you need?

by Tim Koomen

IQUIP Informatica B.V.
PO Box 263, 1110 AG Diemen,
The Netherlands

phone: +31 20 660 66 00
fax: +31 20 695 32 98

email: t.koomen@iquip.nl

Abstract

In this paper the differences between e-business testing and traditional ways of testing are analysed. These differences are caused by the kinds of risks involved in implementing e-business. Specific topics of interest are the kind of expertise you need to have available to the test team and how you can deal with the dilemma between more risks and less time to test.

Key Points:

- Risks involved with e-business are different from traditional software;
- To manage these risks, you have to apply not more but different ways of testing;
- You will require a different mix of expertise in the testing team.

Introduction

The internet can be described as a collection of computers that communicate with each other over a telecommunications network. This network covers the whole world. During the last few years there's been an uncontrolled explosion of internet use, and it seems that this current growth rate is only likely to increase in the coming years. However, there is a visible change in the way the internet is used. Organisations no longer only publish information about their business on the web; they develop more complex applications that make use of the possibilities that the internet offers. This new generation of applications offers new possibilities, but at the same time more risks for business development. This explains the increasing demand for testing because: *no risk, no test*.

This document looks at the testing of internet applications. The emphasis is on applications that offer the most risk and therefore applications where testing is of most importance: e-business.

The document is aimed at people who are currently or shall in the future be involved with the testing of e-business applications. It is assumed that the reader is familiar with structured testing.

Whilst writing this document I have called upon the support and knowledge of many and especially Rob Kuijt, Rob Baarda and Christiaan Hoos, whose written and collected materials I have made use of. Also David Barnett did a great job in translating the document into an English version. I would like to take this opportunity to thank them for all their help.

August 2000, Tim Koomen

Contents

1 The internet

In this chapter we look at aspects of internet applications that are important for a better understanding of the following chapters. The aim is to explain just enough for testing, not to explain all aspect of the internet.

1.1 Application types

Internet fulfills a versatile role in a growing number of situations. Internet applications are used for: information delivery, marketing, entertainment, distribution, image-building, public relations, relationship development, customer service, sales, interaction/feedback with customers and internal communications, etc.

The type of application is important when testing an internet application, as this determines the level of risk.

A company doesn't run much risk when it only publishes a so-called homepage with name and address details. If it is possible for customers to buy (and pay for) articles from the company on the internet site, the risks are much higher. Such applications are called e-commerce applications. E-commerce means "the buying and selling of goods and services or the transfer of money over the Internet, especially the World Wide Web" [whatis.techtarget.com].

An extension of e-commerce is e-business, that as well as sales transactions also includes servicing customers and collaborating with business partners [whatis.techtarget.com]. From here on, only the term e-business is used.

E-business applications can be split into two types:

- Business-to-consumer (B-to-C, also known as B2C)

  This type is used by individual consumers. Examples of this type of application are the sale of books, CD’s or PC’s. Usually it's not known beforehand who the consumers are.
- Business-to-business (B-to-B, also known as B2B)

  This type is used for doing business with other companies. According to Forrester the turnover generated from B-to-B applications will grow explosively in the coming years and much faster than that from B-to-C applications.

Furthermore, there are two special forms of internet applications: intranet and extranet.

- Intranet: a network running within an organisation. Therefore, only users of the organisation network can use the internet-based application.
- Extranet: the organisation network is available via the internet, but only to users whose identity is verified. Extranet users consist of staff and/or business partners of the organisation, and is therefore restricted, unlike the internet.

Both cases are simplified versions of the "normal" internet, because the type and number of users is known beforehand.

.

1.2 Architecture

The (technical) architecture of an internet application is a form of client/server architecture. The figure below shows the components of this architecture.

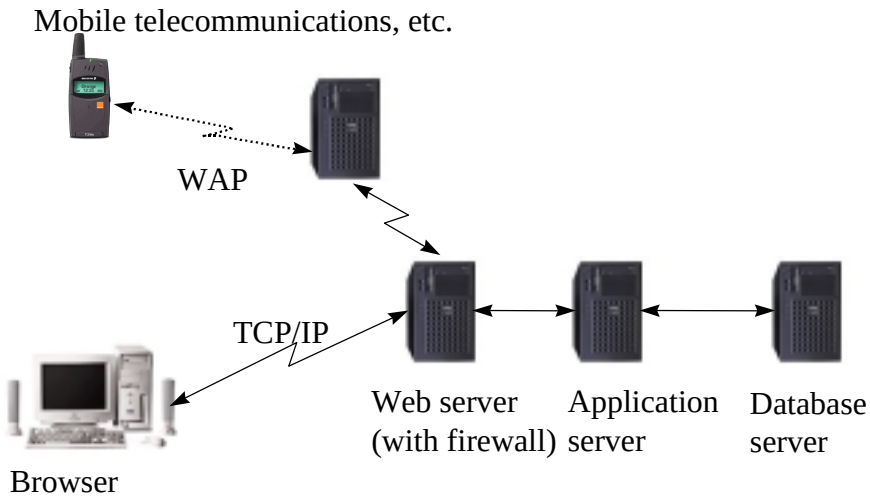


Figure 1, Internet architecture

The user controls the application via a **browser** on a PC, a mobile telephone, TV or another device (the client). A mobile telephone communicates first via the Wireless Application Protocol (WAP) with the server that is connected via a fixed line to the internet. The exchange between the client and server takes place via a **network connection** (with TCP/IP protocol) allowing communication with the **web server**. Often external connection is protected via a **firewall** that is located on the web server. The web server delivers the received information request to the application server (for instance, via the Common Gateway Interface protocol) that in turn communicates with the **database server** in order to query and manipulate the relevant data. These components are not always present. For example a site that only delivers information may only require a web server.

This architecture is characterised by the fact that it comprises a large number of relatively independent components. Many of these components are standard, supplied by a third party. There are estimates that 80 to 90% of the code for an average internet application comes from third party software. Additionally this architecture is not fixed, and it is very difficult to keep pace with the changes in functionality. There are browsers from different manufacturers on the market (Microsoft Internet Explorer (IE), Netscape Navigator(NN)), with different versions (IE3, IE4, IE5, NN3, NN4, etc.). In general the application must at least be suitable for the latest browser versions, otherwise large groups of customers will not be able to use it.

Some of the internet application components:

- HTML pages
- Applications that that can run on a web page:
  - applets
  - java scripts
  - plug-in applications
  - cookies

- Applications that run on the server side:
  - cgi-scripts
  - database interfaces
  - logging applications
  - dynamic page generators
  - messaging agents
- Security:
  - firewalls
  - encryption
  - passwords
- The (possible) situation on the client side:
  - browsers: Netscape , Microsoft Internet Explorer, …
  - operating systems: Windows 3.1/95/98/2000, Macintosh, Unix, …..
  - types of connection: MODEM, ISDN, LAN,…
  - varying connection speed

E-business architecture can also be seen at a higher level of abstraction, one that shows its relationship with other information systems in the organisation, the so called "back-office systems". These information systems are meant solely for internal use and have no direct contact with the customers.

E-business systems and Voice Response systems facilitate new communication channels that have direct contact with the customers. These applications are therefore referred to as front-office applications. However, these applications still make use of the traditional back-office information systems. This makes another type of architecture necessary, Multi-Channel Architecture. Back-office applications often have to be updated to a degree so that they can also be used with the new applications. Usually a middle layer ("middleware") is used to make the communication between the front-office and the back-office systems as efficient and flexible as possible.

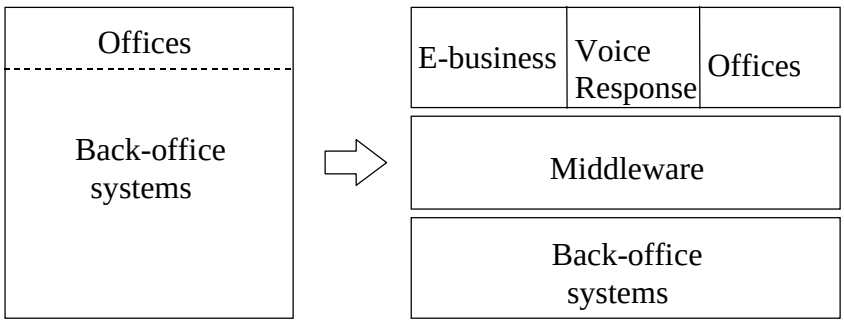


Figure 2, Multi-Channel Architecture

1.3 System development

An organisation, as a first step, usually develops a site for the distribution of company information. The client is not an IT department, but more likely to be the marketing department. The most important requirement is that the site must visually look good.

As a consequence, these first developers aren't often professional developers. They are more likely to be people with a more creative background, for example advertising staff and graphical artists. The fact that these people concentrate more on the appearance of the application than on whether it functions properly, isn't such a problem for these relatively simple sites.

The increasing professionalism of internet sites, in the form of e-business, means that a site that is visually appealing is not enough. The user of such a site expects that it functions and performs well and that financial and private (sensitive) transactions are secure. The high frequency with which updates are released means that the application should be easy to maintain and adapt. Therefore, the demand for professional "software engineers" to develop technically complex e-business applications is increasing.

The high update frequency combined with short time-to-market means that it is important that the development period is as short as possible. Usually a project with a short development period is set up in order to get a first release with limited functionality quickly into production. The functionality is extended with the following releases. Due to this kind of system development and maintenance, the use of the traditional waterfall method is not required. Instead, some sort of RAD approach is used. In practice, it seems that all these approaches have two things in common: there's a minimal amount of documentation and many ad hoc decisions are taken. This of course generates a conflict of interest with the importance of a well designed and maintainable application.

2 More, less or another way of testing?

This chapter discusses the degree to which the testing approach is different for an e-business application. This can be split into two parts:

- How does an e-business application differ from a traditional application?
- What effect do these differences have on the test method?

2.1 What is different?

The most important generic differences between an e-business and a traditional custom made application are given below:

- Time-to-market is always important, however for e-business applications even more so. The necessary time needed to get an application into production is no longer measured in years or months, but in weeks and even days;
- For the majority of organisations e-business applications are new and therefore it also is a new communications channel for customer and supplier contact. The introduction has a far-reaching effect on the existing procedures of an organisation;
- Often the department that has led the development of an e-business application is not, as usual, the IT department; for example it can be led from marketing or sales. Usually there is little IT knowledge in these departments;
- Higher demands are made of aspects such as security, usability and performance. Specific knowledge is needed to reach this higher level of quality;
- The system documentation is often of a lower quality and quantity:
- During the development of an application a large number of standard components are used. Development involves much more integration than programming;
- The users of e-business applications, unlike the IT systems of the internal organisation, are outside of the direct influence of the organisation. There is also little known about the users;
- As e-business applications are aimed at customers and the suppliers of an organisation and thus the primary business processes, the damage, due to insufficient quality is much higher than when an application is used internally. With this in mind, it is often said that a competitor is only one click away;
- Maintenance requires a continuous effort and investment, because the application is never finished and the functionality is constantly being extended;
- The internet changes rapidly. A large degree of change is due to the environment. New components are hastily introduced and are often unstable.

These differences effect the test process in many ways; from the test specification techniques used to the choice of test tools. Risk analysis plays an important role here. A risk is the estimated chance of failure in relation to the expected damage. Most of the differences described above can be one way or another translated into risks. These risks have consequences for the quality attributes to be tested. These

risks have consequences for what to test, and how thoroughly to test. For instance, testing the security or performance aspects is much more important than for traditional applications.

In order to explain the consequences of these differences for the testing process, it is necessary for the reader to have some basic understanding about structured testing. For some additional information on this topic, refer to appendix B, "Structured testing and TMap®".

With regard to the elements of structured testing, the aspects specific to e-business applications are looked at in more detail in the following paragraphs. The following will be discussed:
- quality attributes,
- test levels,
- test basis,
- life cycle model,
- techniques,
- infrastructure,
- organisation.

The last paragraph discusses testing during the exploitation and maintenance phases of an e-business application.

2.2 Quality attributes

The overview below lists the quality attributes (ISO-9126) that demand more attention when testing an e-business application than when testing a traditional application. This is a generic overview and is not meant to replace the test strategy: whether or not to test a quality attribute, and how thoroughly to test, is always a decision to be made for each particular situation.

As an example it should be realised that the importance applied to the quality attributes during testing of intranet or extranet applications normally is very much different to that when testing an open internet application. Intranet applications are not generally used for financial transactions. Therefore the potential damage in case of failure is much less. The number and type of user for intranet and extranet applications is known beforehand, therefore:

- application overload can be more easily prevented (refer to the quality attribute reliability),
- the users can be trained beforehand (refer to usability),
- security is easier to organise and the risks of insufficient security are less.

Quality attributes of higher importance
Security
Usability
Reliability
Time behaviour
Portability
Suitability
Operability (system)
Maintainability

The rest of this section explains the reasons why these attributes are important

+ **Security**

  Security is often one of the first causes for concern. Connection with the internet means that the organisation is vulnerable to undesired intrusion from outside. Furthermore, information that is exchanged is vulnerable to interception or manipulation. How sure is the customer that his credit card payment is not intercepted and misused? Generally the following kinds of intrusion can be distinguished:

  - Unqualified entry

    The system (program, data, network traffic, etc.) is entered by people that don't have the necessary rights to do this. No changes are made, however information can be stolen. This type of intrusion can come from an internal (own staff) or an external (hackers) source, and targets the infrastructure of the organisation or the user, or the communication between the two.
  - Changing information/pages

    A more severe form of intrusion is when intruders actually change information, programs or web pages.
  - Affecting availability (denial of service)

    The performance of a system can be influenced so that it is effectively no longer available. An example of this is the E-mail bomb; a bulky message that is used to overload the handling capacity of the recipient's post-box.
  - Viruses

    Programs or text (via macros) can contain viruses. Normally, a virus first spreads itself and at a later stage becomes visible. As well as infecting all of the programs on the PC of the recipient, the virus can also spread itself via the network or via mail (by automatically sending itself to all of the addresses in the address list a chain reaction occurs). When a virus exposes itself it can unexpectedly change text or even delete data from the PC.

  Precautions against such forms of intrusion are the use of firewalls, virus detection software, logging and secured transactions (SSL, secure socket layer, and SET, secure electronic transaction) and mail (PGP, pretty good privacy). It must be noted that security is a continuous technological struggle between intruders and system administrators. A system that today is optimally secured, can become a security worry tomorrow.

  The security risk posed by intranet applications is less because there is no connection with the outside world. Also, this sort of application is used to disperse information rather than facilitate financial transactions where the potential damage is highest.

+ **Usability**

  The fact that an unknown internet user should be able to use the application without training demands more user-friendliness than a traditional application, especially when the user can just as easily do business with a competitor. Aspects that play a part in usability are learnability, operability, comprehensibility and clarity. The following aspects are also important:

  - Attractiveness

    Is the application designed well; does it look attractive and logical?
  - Accessibility

    Can a user find the application easily? Also, if the internet address is not known and it must be found via a search engine?

Another way of looking at usability is in the way the application helps the user. Does the application have added value for the user? For example, does it simplify the sales process? An example is the possibility to search for the required item in different ways. Another is if you find a book or CD interesting, then the site recommends other books or CD's that you might also be interested in. A further step is that the application stores information about the customer. The application can then supply the information that best suits this customer profile. The usability aspect is crucial for a website's success according to a number of parties (Patricia Seyboldt, Forrester).

+ **Reliability**

  Reliability is considered to be a very important quality attribute for e-business applications. Increasing numbers of users have often overloaded applications, rendering the application no longer available. The risks are relatively high as it is difficult to estimate the number of users beforehand, the application must be available night and day, and failure of the application means a loss of earnings. It is often quoted that due to failure 20% of the transactions are lost and that the remaining 80% are completed later. In the United States Forrester Research did some research in 1999 on the topic of loss of revenu when a site is down for one hour: for Amazon this is estimated at $22,500 at daily revenues of $ 2,7 million and for Intel $274,980 at daily revenues of $ 33 million. An important element of the required reliability is the scalability: the ability to change the number of servers being used, dependent upon the load. A term related to this is "webfarm". A webfarm is a group of servers that are located together in one location. The number of active servers depends upon the load. Failure of a server can be recovered by activating one of the other servers.

+ **Time behaviour**

  This quality attribute is linked strongly with reliability. Will the performance of the application be acceptable? Currently the 8-second rule is considered to be of acceptable level. Above this level the users get impatient and irritated. It should be noted that overall performance is dependent upon a number of external elements that the organisation can't control: the computer and modem of the user, the telephone network and the infrastructure of the provider. The performance of an application can also be compared to that of competitor sites.

+ **Portability**

  The large diversity of components that comprise an e-business application means that this quality attribute is more important than normal. Does the application work with an old browser version? Does it work with a new version? If the user does not accept cookies, can he or she still use the application? If not, should the users be informed? A confusing factor for the portability quality attribute is that there are actually standards defined (for example for HTML), but usually these standards are not adhered to. For example, the browsers from Microsoft and Netscape contain all sorts of additional functionality not included in the HMTL standard.

+ **Suitability**

  At the moment e-business applications represent a new way of working for most organisations. This means that the organisation must set up staff and procedures to work with the application. It is unsatisfactory when handling of incoming requests takes days. One of the most difficult aspects is keeping the information shown on the site up-to-date and accurate. A four months old "new CD of the week" is an example of the opposite.

+ **Operability (system)**

  A number of aspects make operating the application for system managers more complex than normal:

- the large number of components;
- the supplied (system administration) documentation is often insufficient.

\+ **Maintainability**

A feature of e-business applications is that they must be changed quickly and often. Precautions to make the application as maintenance-friendly as possible are for example, good documentation (including test documentation), and the use of HTML-pages dynamically generated from a database instead of static pages.

Probably the most important attribute for testing, accuracy, "does it do what it is supposed to do", is not classed as one of the quality attributes requiring more attention than usual. However, testing the accuracy is still very important and still demands a large part of the test effort.

= **Accuracy**

When an average e-business application doesn't function correctly it usually causes more damage than a traditional application. However, the risk is also defined by the chance of failure. Currently, the functionality of e-business applications is not regarded as extensive or complex. There is also less chance of failure (however still present) as many standard components are used during development. The higher degree of damage is therefore compensated by the lower chance of failure so that the testing of functionality does not demand any more attention than normal.

As so many standard components are used, testing should take care not to test these components extensively. Just as when testing an ERP package one should ask "what doesn't need to be tested?". For instance, user acceptance testing of the Netscape browser is a bit of a waste of time. Knowledge of the test object is indispensable. As the number of components and combinations of components is numerous, the integration of (standard) components should be an important test objective. Attention should be paid to the fact that standard components contain a certain degree of instability. During testing this results in large number of non-repeatable errors. As it is not repeatable the cause of the error can not be analysed and therefore not fixed. A well-known example of a non-repeatable error is the locking of a browser whilst working with an application. It is much easier to find the cause of non-repeatable errors during white-box testing (especially the integration testing done by developers) when only a few components are tested together than later during black-box testing when the complete application is tested.

2.3 Test levels

While testing e-business applications we can use the traditional test level classification:

- program test;
- integration test;
- system test;
- acceptance test.

With the addition of a new test level:

- production test.

During production the application is tested to check that it still functions correctly and that the performance hasn't decreased. The reason for this level of test is that there are all sorts of elements (number of users, browser, provider, bandwidth, external links) that the organisation has no control over. If not tested, a poor

response from a provider or a link that does not work any longer remains unnoticed. Another cause of performance degradation can be due to the increased use of the application. This is very difficult to predict beforehand.

Along with testing, evaluation activities like reviews and inspections are important elements of quality control for an e-business application. All of the different test aspects should be covered by one or more of the test or evaluation levels.

The co-ordination of all of the different levels of test plus evaluation is very important when trying to develop an optimal test strategy: to find the most important errors as early and as cheaply as possible. To achieve this, prepare (and monitor) a master test plan (often also called a Validation and Verification Plan). The advantage of a master test plan is that it makes a co-ordinated overview of the test process possible, and also makes it possible to share scarce knowledge and resources optimally over the different test levels.

2.4 Test basis

Testing is basically comparing. Based on defined input, the actual output of the software is compared with the expected output and differences found are called defects. In order to determine the expected output, a point of reference, for example documentation, is needed. This is called the test basis. All references from which the requirements of an information system can be inferred, form the test basis. From this test basis, test cases are derived.

A RAD type of approach is normally used during systems development for e-business applications. The functionality of the application is determined in workshops by the developers and users. The quantity and quality of the system documentation is often minimal.

Possible forms of test basis are:

- Requirements documentation

  In these documents the objectives and requirements to be fulfilled by the application are defined. This documentation is essential as there's not a lot of attention paid to the functional and technical specifications. Business scenarios and use cases are often part of this documentation.
- Functional and technical specifications

  This form of test basis is usually less extensive for internet applications than for traditional applications
- Norms and standards

  As well as norms and standards defined for the project, existing norms and standards of the organisation or even external norms and standards can be used. An example of the latter is the 8-second performance rule.
- External references

  A competitor site is a very useful test basis!

If the documentation for a test basis is incomplete then the testers can usually gather complementary information through interviews, attendance of workshops, (jointly) defining business scenarios etc. (as such the knowledge of individuals forms the test basis). A sufficient amount of test cases can therefore be defined to provide confidence in the quality of the test. Involving the test team at an early stage in the development process can facilitate the collection of information for a test basis. The test team should advise clearly on the risks involved with an insufficient test basis.

Whilst defining the strategy it can be agreed not to test but to measure some aspects due to a lack of requirements. The result of such a measurement can be that a maximum of 100 users can work with the application at the same time before the performance begins to degrade exponentially. Testing is more than measuring because it also expresses whether measured results are satisfactory or not. For instance, the resulting 100 users measured is less than the required 500 users, therefore the application will not perform as required.

2.5 Life cycle model

All structured testing approaches make use of a life cycle model for their activities. A simple model consists of three phases, planning, specification and execution. More mature models offer five phases: planning, preparation, specification, execution and completion. These models remain valid for e-business testing. The only point of attention is that acquiring a usable test basis for e-business applications demands more time than usual. Refer to the previous paragraph for further explanation.

2.6 Techniques

The quality attributes to test and the type of test basis have a large influence on the choice of test specification techniques. In practice this results in the choice of informal test specification techniques like error guessing.

The reuse of existing testware is an important option when developing test cases. An e-business application often contains the same functionality as the existing traditional office application, so testware of this application can perhaps be reused.

Many checklists, derived from practical experiences, are used.

Below some specific hints and directions for testing certain quality attributes are given:

- Usability

  For the testing of this quality attribute, techniques such as eye-catching are available. These tests are usually performed in a usability lab because of the specific infrastructure required.

  Checklists such as SUMI (Software Usability Measurement Inventory) or WAMMI (Web Analysis and Measurement Inventory) are also used.
- Security

  If actual testing takes place, this is called penetration testing. In this form, ethical hackers attempt to break into the application. They have the tools and the knowledge to test the security thoroughly. Also security audits are widely used.
- Reliability and time behaviour

  Load and stress testing is the common term for testing the attributes reliability and performance. The system is tested to see if it remains sufficiently available (and the effect upon performance) when subjected to normal and maximum expected load, and where the bottlenecks are. The situations to be tested are as follows:
  - simulation of one user;
  - simulation of the average number of users expected;

- simulation of the maximum number of users that the system should be able to handle;
- a heavier load than the maximum expected, in order to determine the level of load at which the availability degrades markedly;

The test situations above are based upon the presumption that the availability degrades slowly up until a certain level of loading, but thereafter degrades at a quicker rate. Refer to the diagram below.

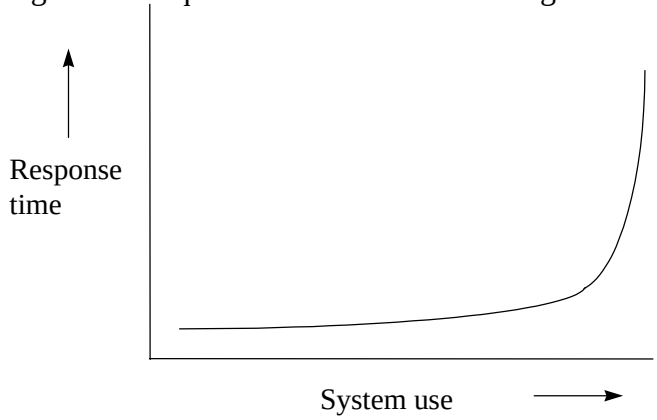


Figure 3, Degradation curve

At which point(s) is the degradation to be measured? An e-business application comprises a large number of components. A first variant is to measure the response over the complete application. How long does it take before information from the database is displayed to the user? Another variant is to measure phases, for example between user and web server, between web server and application server, between application server and database server. The choice of variant is dependent upon the risks and how difficult it is to analyse the bottlenecks.

For this form of testing the use of test tools (load/stress) is required. Employing a large number of "real" users to attain system loading can be done once with a lot of effort, but often not for a second time. In reality these tests should be repeated periodically as changes are often made to the hardware and software.

- Portability

  During the testing of different browsers, the standard settings (colours, fonts, Java Y/N) are often changed. Constructions that are incorrect such as the hardcoding of settings in the programming are thus quickly detected.

2.7 Infrastructure

2.7.1 Test tools

A large number of test tools, specific for the internet, are available. The suppliers of these tools put a lot of effort into marketing, resulting in the fact that at conferences the testing of internet applications seems to be synonymous with the use of test tools. A reason for the popularity of test tools for testing internet applications is that it's relative simple to automate many of the operations. The number of test tools, some for free, some not, is extensive. Operations where test tools are widely used:

- Spell-checkers;
- Image analysis (size, bandwidth, “how long does it take to load an image?”, ...);

- Structure controls (dead links, loose ends, old parts, ...);
- Portability controls (differing browsers);
- Monitor width controls (does the application fit smaller monitors/settings?);
- External link controls;
- HTML validators;
- Web page syntax and style controls.

As well as the test tools specific to internet applications traditional test tools can also be used, for example:

- Defect administration tools
- Record & Playback;

  As e-business applications often and quickly have to be updated, a good regression test is very important. The automation of this test is recommended, especially when you consider the later tests (during maintenance) when regression tests become even more important.

  IQUIP uses a structured approach called TAKT to set up these tests properly.
- Load & Stress;

  These tools generate load by simulating a large number of users in order to test whether the system still functions properly and quickly enough under the expected production load. These tools often have monitoring functionality to measure the results. For the measurement of reliability and time behaviour, these tools are virtually a necessity.
- Monitoring tools;

  In order to get an indication of aspects such as memory requirements, CPU use, network strain and performance, monitoring tools can be used. All sorts of data about resource usage are measured and recorded. Tools, specifically for the internet, are available that periodically measure the availability and performance of a live production application.

2.7.2 Test environment

Often during the development of the first generation of internet applications there was only one environment: the production environment. The user was often informed that the site was under construction. This is not acceptable for e-business applications; there is usually a separate development environment. However, a separate environment for testing is not usual. It is expected that this will change in the nearby future due to the increasing importance of e-business applications and the testing of them. Reliability and time behaviour testing are the main factors that make a separate environment necessary.

The large number of components used makes the installation and management of the (test) environment a complex matter.

2.8 Organisation

2.8.1 Knowledge

The most important organisational difference between e-business applications and traditional applications is the knowledge required in the testing team. Knowledge can be split into the following sub-categories:

- System

As a large number of standard components are used, it is important during testing to check that this standard functionality is not tested. This demands knowledge of the internal structure and functionality of the test object. A developer is more likely to have this knowledge than a tester. Insufficient knowledge of the architecture of the internet application leads to time lost on tests that aren't required.

- Testing
  As well as generic test knowledge (such as TMap), additional specialist knowledge (dependent upon the quality attributes to be tested) is needed. Examples of this are the testing of security, usability and user-friendliness.
- Test tools
  Refer also to paragraph 2.7.1. To use a test tool, knowledge of the tool is required. The array of test tools for internet applications means that the test team must acquire knowledge of the different tools to use.
- Subject matter and organisation
  As the system documentation is often insufficient, knowledge of what the application is supposed to do, is an alternative form of test basis.

Different (often specialist) kinds of knowledge are required. The test manager needs good organisational skills in order to bring together multi-talented test teams with the right composition of knowledge at the right time.

2.8.2 Organisational forms

The different test levels referred to in paragraph 2.3 need to be adjusted to correspond with the complete system development process. The most important question that can be asked is if the test level is organised as an independent process or as part of another process. The organisation is always dependent upon the situation. However in the table below preferences are shown.

Test level	**Organisational form**
Program test	Part of the development process.
Integration test	Part of the development process.
System test	Part of the development process. Name a test leader who only leads and/or performs test activities (and no development activities). This is done to prevent the test from being snowed under by the rest of the development process.
Acceptance test	Independent; can be split unto a user acceptance test and a production acceptance test. It is possible to contract out some of the activities (for example testing the usability or the security).
Master test plan	The above test levels can best be co-ordinated centrally by a test manager or test co-ordinator.
Production test	Part of the system management process. Can be contracted out.

2.8.3 Parties involved

E-business applications are often initiated by non-IT departments such as purchasing, sales, marketing or support. Often the application has impact on other departments. It is important to involve these departments for their knowledge of the subject and the organisation; this improves the quality of the application and therefore improves the degree of acceptance.

It should be understood that people from these departments often have little knowledge of IT, with comments such as "testing isn't necessary as the developer

has enough knowledge of the subject". Some test awareness activities are often necessary to make clear the importance of testing and what it (globally) involves.

2.9 Operation and maintenance

At a certain moment the application has been developed and is in production. For traditional applications a reasonably stable situation has been attained: most changes to the system and infrastructure are manageable and can be planned easily, and the use is predictable. This isn't the case with e-business applications: many changes can't be managed and the application is much less predictable. Links to other sites seem suddenly no longer valid, the search engines no longer work effectively, newer versions of browsers and system software are released and use of the system multiplies from one day to the next.

This means that more maintenance (testing) takes place more often, and therefore it's necessary to perform a regression test in order to ascertain if the complete system still works satisfactorily. This test not only validates the accuracy of the application, but also for example the portability and time behaviour. A consideration is to perform this test periodically, regardless whether or not a change has been made or not, as part of the so-called production test (refer to 2.3 Test levels). The high frequency with which this test is performed over and over again justifies its automation. This test automation is best started during the initial development phase as most knowledge is available at this point and the automation can be used for a longer period. Thus more time is available to earn back the investment.

3 The future of e-business testing

As with e-business in general, there are also a number of trends witnessed in the testing of e-business applications that are a result of the increase of e-business and the growing competition:

- The development of e-business applications is increasingly performed in a professional way. This results in an increased awareness of the importance of testing. This, in part, means that testing in the future will be performed in a more traditional way than currently.
- The growing importance of e-business means also that there is a switch of some of the effort from development to testing. This in turn demands increased technical knowledge for testing.

This trend is shown in the figure below.

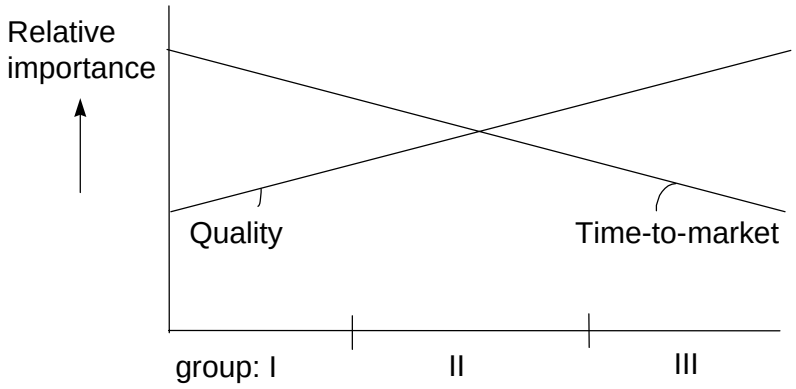


Figure 4, E-organisation classification

Generally, there are three types of organisation (shown from left to right) that can be defined:

I This group represents the newcomers to the e-business market. These companies often have a product or a service that clearly differentiates them from competitors. To take advantage of this temporary lead, time-to-market is extremely important. The quality of the e-business application is therefore less important. The organisation relies upon the differentiating power of the product or service. The recent battle between the free internet providers is an example of this group.

II The organisations in this group have gained some experience developing e-business applications. They usually don't have much experience with IT for support of the primary business processes. Although time-to-market is still very important, the quality of the e-business application is also important. However, the role that testing plays is still unclear. Organisations belonging to this group operate in industry, the trading world, publishing and governmental services. Companies for which e-business is the primary business (the so-called "dotcom" companies) also belong in this group.

III This represents the group of companies where IT has supported the primary business processes for a longer time. Time-to-market remains as always important, however the (direct and indirect) potential damage that can result from insufficient quality of the application in production is very high, and therefore the quality aspect is very important. There's a belief that structured testing is important in order to determine the level of quality of the application. Organisations from the world of banking, insurance,

telecommunications, some governmental services and a few "dot.com" companies are examples from this group.

Appendix A, References

- Pol, M., R. Teunissen, E. van Veenendaal (1999), *Testen volgens TMap*, Tutein Nolthenius, 's-Hertogenbosch, ISBN 90-72194-58-6
- Koomen, T., Pol, M. (1999), "Test Process Improvement: a practical step-by-step guide to structured testing" Addison Wesley, ISBN 0-201-59624-5
- Samaroo, A., S. Allott, B. Hambling (1999), *E-ffective Testing for E-business*, Imago^{QA}
- Bazzana, G., E. Fagnoni (1999), *Testing Web-based applications*, SQE Europe (http://net.onion.it/)
- Boreel, M., *Internet Tussen Hoop en Vrees*, IP/Vint, ISBN 90-75414-04-8
- Whatis?com, http://whatis.techtarget.com

Appendix B, Structured testing with TMap®

This paper provides some background information on structured testing in general and more specific on IQUIP's structured testing methodology TMap, the Test Management approach. In recent years TMap has evolved towards the standard for software testing in The Netherlands. It represents an extensive collection of standards, instruments, tools and procedures which may be selected for a specific test, using the test strategy development technique. Most Dutch banks, insurance companies, pensions funds and government departments use TMap partly or as a whole. In fact, more than two hundred Dutch organisations use TMap. The TMap book (in Dutch) has proven to be a best seller, international interest and awareness has resulted in an English introductory book. A full translation is expected to be published in spring 2001.

Any structured testing approach seeks to provide answers to the what, when, how, where and who questions of testing. To structure the organisation and execute the test processes TMap is based on four cornerstones:

- a development process associated Life cycle model for the testing activities (L);
- usable Techniques for performing the testing activities (T);
- the right Infrastructure and tools (I);
- sound Organisational embedding (O).

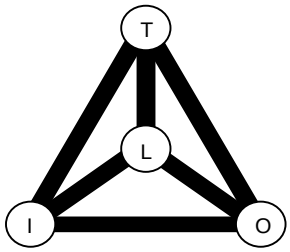


Figure 1 : the four cornerstones for structured testing

Life cycle

Like a system development process, a test process consists of a number of different activities. There are life cycle models for system development, which make it possible for the various activities, their sequence and interdependence to be mapped out. In such a model, the entire process is divided into a number of phases which, in turn, are subdivided into activities. The objective, input, process, output, tools, etc. of each activity are described in detail. The use of a life cycle model makes it possible to retain an overview during the system development process. There are also life cycle models for the test process itself, and they are essential for retaining an overview during testing. The life cycle model is like a thread running through the process. By describing what, when, how, with what, where, by whom and so on something needs to be done in the course of the test process, demands on, and relations with, the remaining cornerstones (techniques, infrastructure and organisation) are automatically established. It is valid to say that a life cycle model may be applied to any type of test. A good test life cycle model is therefore the first cornerstone supporting a structured test process.

The life cycle model for testing operates in parallel with the life cycle models for system development. In the TMap life cycle model the three main testing activities are divided into five phases. In addition to the planning and control, preparation, specification and test execution phase, a completion phase has been defined to finish the testing process in a structured way and to preserve the testware for the first or next maintenance release.

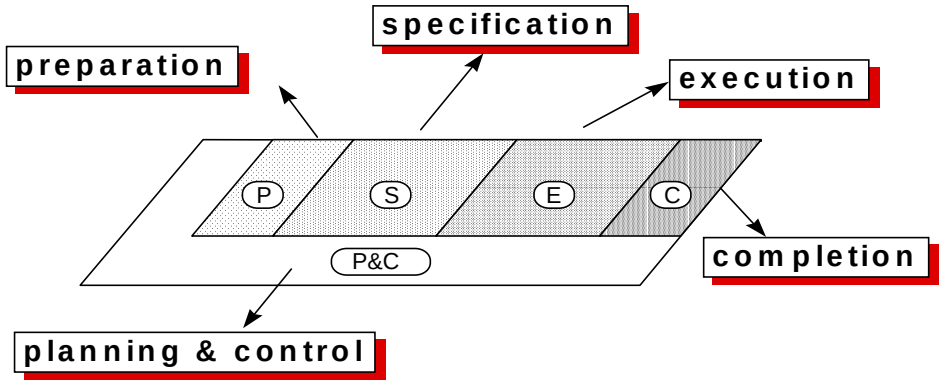


Figure 3: The TMap life cycle model

Techniques

Many techniques are available for supporting testing. There are techniques for supporting the planning process and risk based testing as well as techniques for study and intake of the test basis, and reporting techniques. Test specification techniques constitute the most important group and vary from very informal to very formal (mathematical). A test specification technique may be used to measure certain characteristics of an information system, or quality characteristics as they are called. However, there is not a single test specification technique that can measure all characteristics, and it is therefore necessary to use a combination of techniques for this purpose. A 'toolbox' well-stocked with test techniques is the second cornerstone supporting a structured test process.

Infrastructure

A 'test environment' is needed if tests are to be executed. Such an environment must be stable, controllable and representative. It must also be separated from other environments (the development environment, for example). It is only under the above conditions that reproducible tests can be carried out. If this is to be done efficiently, certain aids - test tools - will be required. The office environment of the test team is also important. A good infrastructure is therefore the third cornerstone supporting a structured approach to testing.

Organisation

A test process is performed by people and therefore it requires organisation - on the one hand organisation within the test team whereby everyone must be given tasks and responsibilities, and, on the other, incorporation of the test team in the project or line organisation. It is essential to specify for each test who is to prepare and execute it, who is in charge, who is the one to check the quality, and to whom the results are to be communicated. It is only under these conditions that the test process can proceed in an orderly fashion. A good test organisation is therefore the fourth cornerstone supporting a structured approach to testing.

Thursday 7 December 2000

T1

e-Testing: What e-Xpertise Do You Need?

Tim Koomen

Graduated in 1986 in Informatica at the University of Amsterdam, since 1992 Tim Koomen has been a professional tester and has performed most testing functions since then. He participated in several testing projects for clients of IQUIP Informatica in the Netherlands, a company with over 300 dedicated testers and owner of the structured testing approach TMap^{Ò}. Tim is a member of the R&D-team, covering issues like testing Component Based Development, Test Process Improvement and testing E-business. For IQUIP, testing E-business software is rapidly becoming a large part of our business. Tim is co-ordinating the testing efforts for multiple clients of IQUIP.
He is the co-author of the TPI-book, translated in Dutch, English and German, and frequently presents at conferences (Eurostar '97, Eurostar '98, Quality Week Europe ’99: TPI tutorial and two presentations, Eurostar ’99: discussion session and presentation, tutorials at ICS Test 2000, SQE Congress 2000, Quality Week 2000) and training sessions throughout Europe and the United States.