Code Security Reviews durchführen

Ein Code Review ist nach gewissen Qualitätsregeln (Audits) und Metriken durchführbar (Struktur vor Funktion). Ein Security Review basiert auf Stabilität, Robustheit und Güte der Algorithmen.

Kleiner
Kommunikation.....

EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Code Review Ziel

- Das Ziel eines Code Review besteht darin, ein klares **Verständnis** darüber zu erhalten, wie ein Software-System strukturiert ist, welche Protokolle, Formate und Schnittstellen anhand der Architektur vorgesehen sind; Eine Architektur basiert auf: Plattform, Framework, Topologie
- Ein Code Review prüft und beurteilt die nichtfunktionalen Anforderungen, Qualität (Fehlerminimum) bezüglich Architektur und Substanzwert !
- Eine Inspektion spezialisiert sich auf die Fehlerfindung, nicht ihrer Beseitigung. Generelle Reviews prüfen auch den Projektfortschritt.

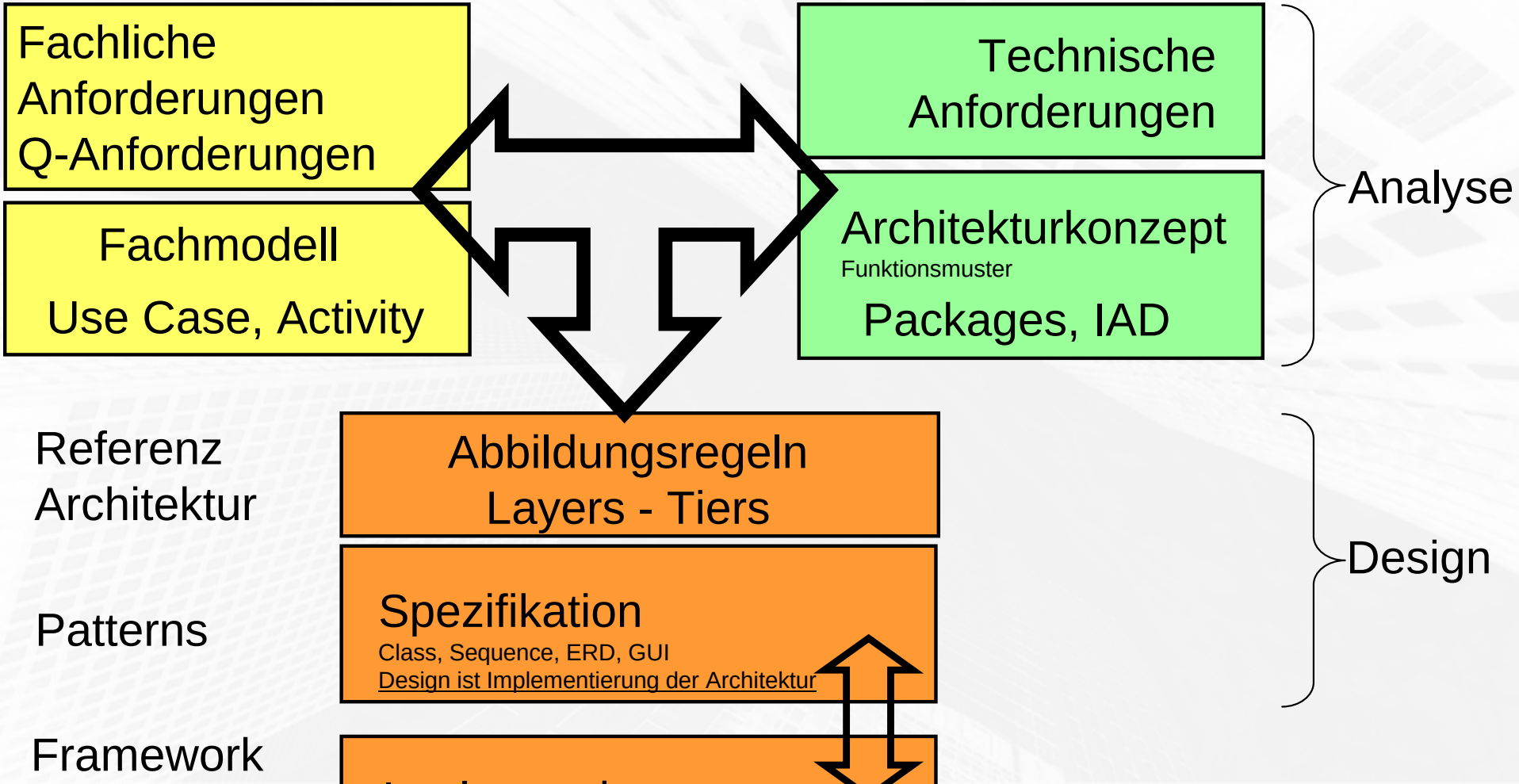
EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Code Review Ziel II

- Zu schaffen sei ein IT-Gütesiegel, vorab bspw. durch die OMG zertifiziert (Technical Engineering Expert, Technical Excellent Engineering) or something:

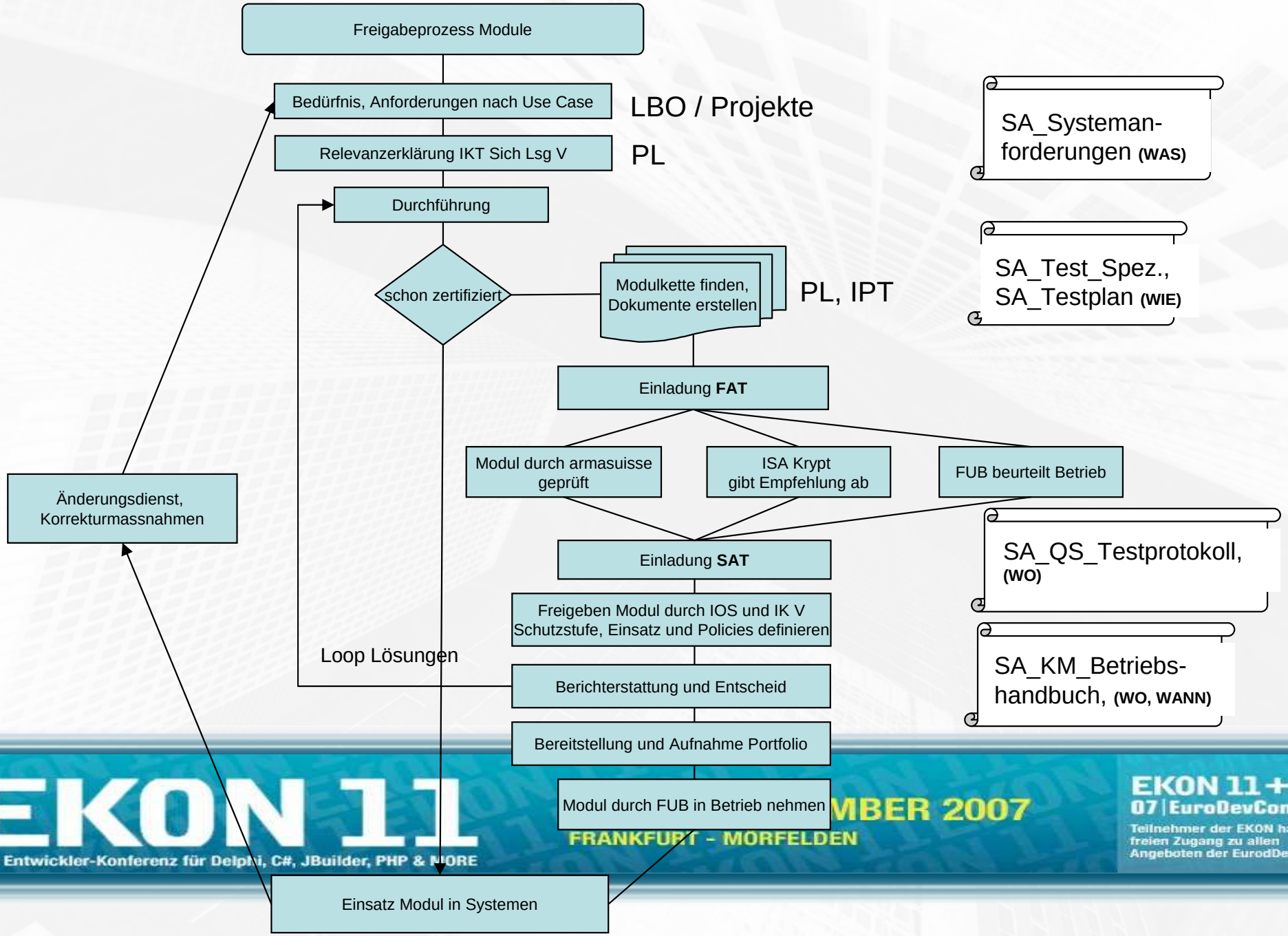
EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Rahmenbedingung



EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
MBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Prozessmodell Zertifizierung Sicherheitsmodul



EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
MBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Nichtfunktionale Anforderungen

- Modularisierung
- Erweiterbarkeit, Flexibilität
- Wartbarkeit, Wiederverwendung
- Testbarkeit der Spezifikation
- Stabilität und Robustheit

Sekundäre Anforderungen: (Wechselwirkungen)

- Skalierbarkeit
- Redundanz und Verfügbarkeit
- Portabilität, Mehrsprachigkeit

EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Minimale Anforderungen

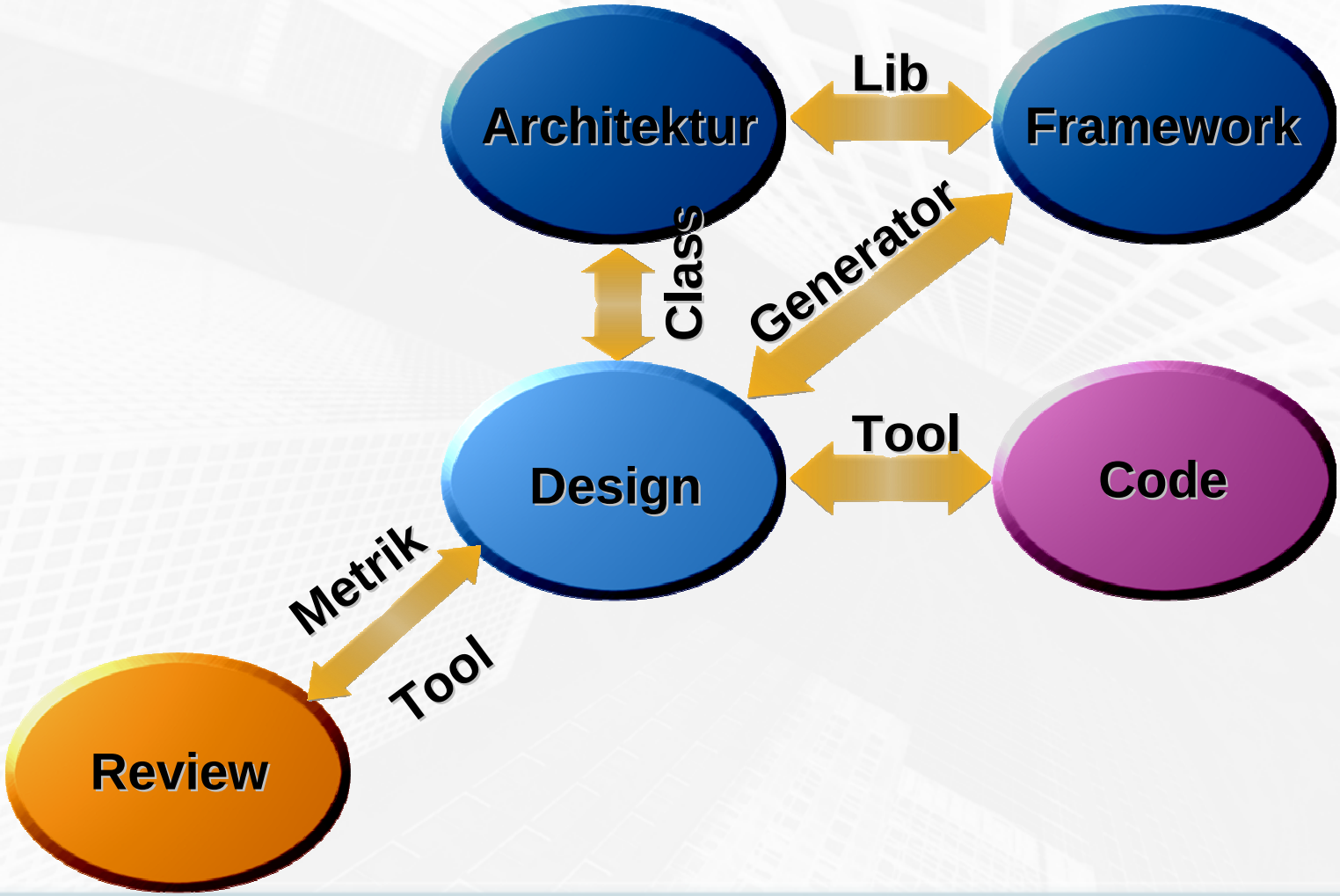
1) SW ist durch das Betriebssystem und durch die Applikationen **nicht beeinflussbar** (sofern nicht in einer sicheren Umgebung)
Beispiel: eine externe VPN-Box enthält die überprüfte SW, die die Kommunikation des Netzwerkes schützt

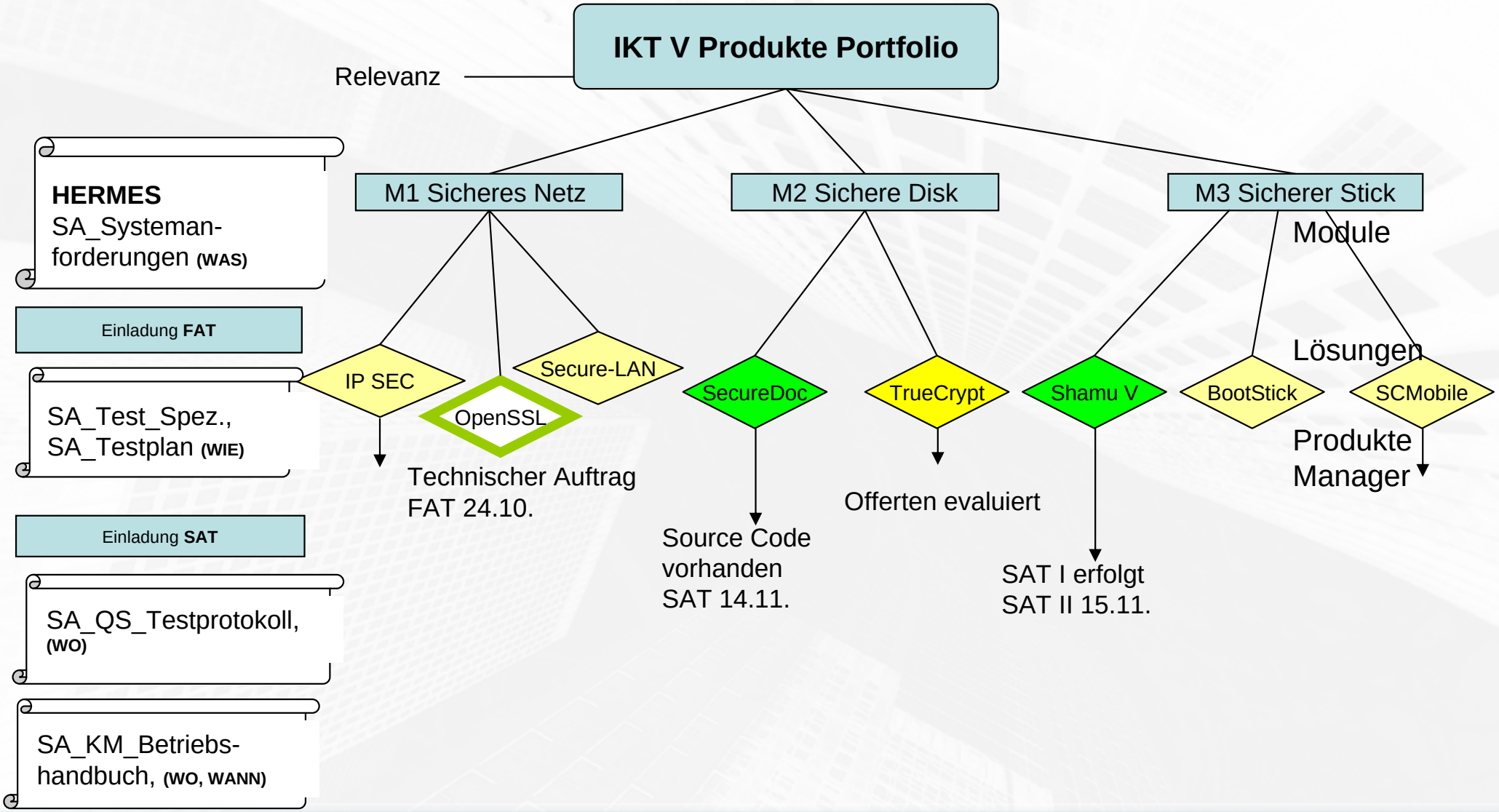
2) SW darf **nicht manipulierbar** sein
(Schutz gegen Viren und Malware)
Beispiel: nur überprüfte und digital signierte ausführbare Programme sind dem Betriebssystem weitergeleitet

3) SW muss **überprüfbar** sein
(Nachvollziehbarkeit)
Beispiel: vollständiger Source Code muss vorhanden sein

EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Review Umfeld




IKT V Produkte Portfolio
Relevanz
HERMES
SA_Systeman-forderungen (WAS)
Einladung FAT
SA_Test_Spez., SA_Testplan (WIE)
Einladung SAT
SA_QS_Testprotokoll, (WO)
SA_KM_Betriebs-handbuch, (WO, WANN)
M1 Sicheres Netz
M2 Sichere Disk
M3 Sicherer Stick
Module
Lösungen
Produkte Manager
IP SEC
Secure-LAN
OpenSSL
SecureDoc
TrueCrypt
Shamu V
BootStick
SCMobile
Technischer Auftrag FAT 24.10.
Source Code vorhanden SAT 14.11.
Offerten evaluiert
SAT I erfolgt SAT II 15.11.

Vorbedingungen

- Dokus, Testklassen zu Algorithmen einsehbar
- Vorabversionen und Updates
- Installation, Support, Hotline der Firma bekannt
- Volle Verfügbarkeit über den Sourcecode
- Compilationsfähigkeit erstellt ($D, $L, tdb32info)
- Benchmarks und Memoryleaks geprüft
- Tools und Review Kriterien vorhanden
- Sprache und Form bei Dokumenten festlegen
- Referenzarchitektur und Patterns bekannt

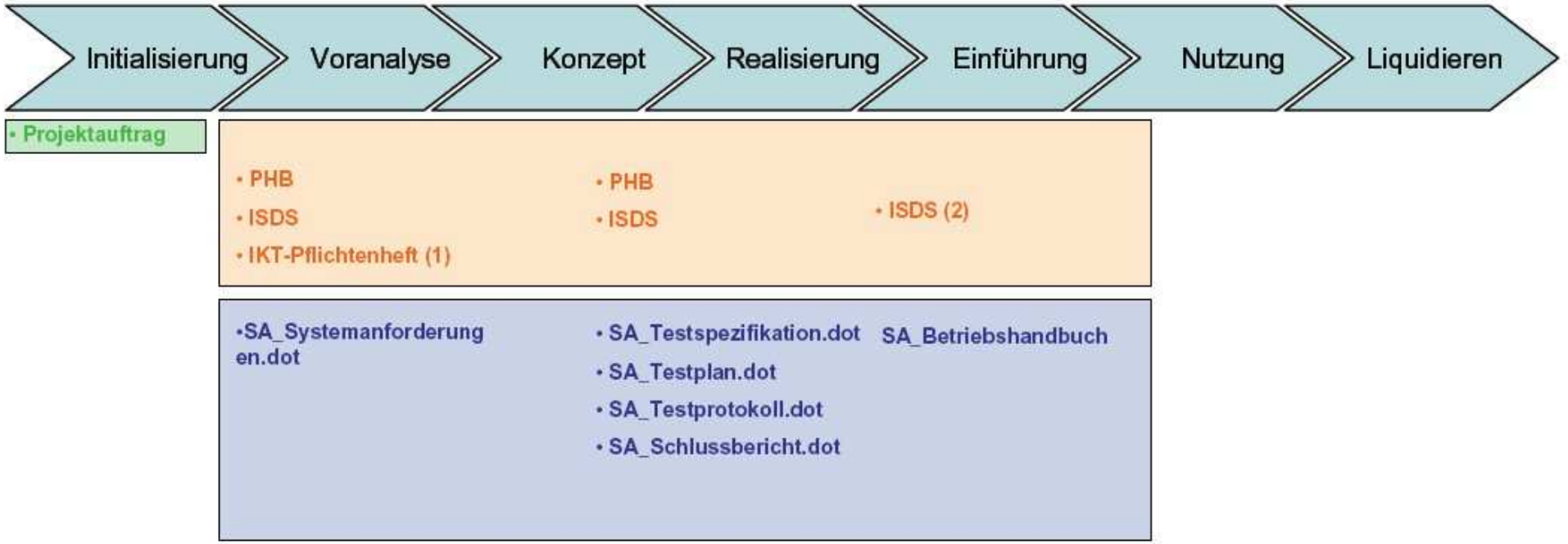
EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Definition Vertraulich

Art. 7 Kategorie VERTRAULICH
Als VERTRAULICH sind Informationen zu klassifizieren, deren Kenntnis durch Unbefugte es ermöglicht, dass:
a) die Regierungstätigkeit schwer beeinträchtigt wird;
b) die Wirksamkeit wichtiger staatlicher Massnahmen vereitelt wird;
c) bedeutende Fabrikations- oder Geschäftsgeheimnisse verletzt werden;
d) eine Strafverfolgung vereitelt wird;
e) die Sicherheit von bedeutenden Infrastrukturanlagen schwer beeinträchtigt wird.

EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Doku Anforderungen



EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Security Leak ?

Mit TSQLConnection Bsp.

```
Connection:= TSQLConnection.Create(NIL);
 with Connection do begin
   ConnectionName:= 'VCLScanner';
   DriverName:= 'INTERBASE';
   LibraryName:= 'dbexpint.dll';
   VendorLib:= 'GDS32.DLL';
   GetDriverFunc:= 'getSQLDriverINTERBASE';
   Params.Add(,user_name=SYSDBA');
   Params.Add('Password=masterkey');
      with TWebModule1.create(NIL) do begin
       getFile_DataBasePath;
       Params.Add(dbPath);
```

Wartbarkeit

- Namenskonvention, Namensräume, Kommentare
- Bridge zwischen Interface und Implementation (siehe auch Modularisierung)
- Installation und Configuration Guide durchsehen
- Abhängigkeiten und Fehlmanipulationen der Objekte klären:

```
for i:= 0 to SourceTable.FieldCount - 1 do
    DestTable.Fields[i].Assign(SourceTable.Fields[i]);
DestTable.Post;
```

EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Wartbarkeit II

- die Dokumentation, insbesondere die exakte Spezifikation von Schnittstellen (Interfaces)
- die lokale Verständlichkeit von Anweisungen resp. von Kommentar im Code, der Parameter
- das Vermeiden von Duplicated-, Dead Code
- in das Programm eingebaute Prüfungen der Annahmen, die man über Zustände hat (Assertions)
- ein möglichst großer Umfang von automatisch ausführbaren Tests für das System (DUnit)

EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Wartbarkeit III

- die Dokumentation, insbesondere die Verständlichkeit von Komponenten

Bsp. Im Gegensatz zu UDP ist TCP verbindungsorientiert. Dieser Unterschied ist durch die Änderung der Methodennamen zu betonen:

UDP: send und receive, receiveListener

TCP: write und read, ReadListener

Da TCP verbindungsorientiert ist, erkennt der Socket auch Verbindungsabbrüche.

EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Compilerdirektiven warten

- Die Direktiven sollten als *.inc file vorhanden sein ({$I IdCompilerDirectivs.inc}, {$I IdVers.inc}).

```
{$H-}
STRING Längenbyte + ARRAY[1..255] OF CHAR;
STRING[16]Längenbyte + ARRAY[1..15] OF CHAR;
{$H+}
STRING Zeiger auf: ARRAY[0..?] OF CHAR + #0;
STRING[16]Längenbyte + ARRAY[1..15] OF CHAR;
```

Testbarkeit (White Box)

- Es lassen sich beim Ändern einer Subroutine schnell mal die davon abhängigen Prozeduren aufzeigen (z.B. Bindings Report).
  Viele haben ja ein eigenes, „automatisiertes“ Verfahren entwickelt, um Code auf Fehler hin zu durchsuchen (Testprotokolle und Userchecks).

  Bsp.: MessageDlg('are you sure to exit?',
  mtConfirmation, [mbOk], 0);

- Wurde das Error Handling, Logger getestet ?
- Sind Testklassen wie DUnit und zusätzlichem Refactoring im Einsatz?

EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Testbarkeit Praxis

```
function letAllowInstances(const mainForm, cmdLine:
                                                PAnsiChar): boolean;
var keyWindow: HWND;
    begin
      result:= false;
      keyWindow:= FindWindow(mainForm, NIL);
      if (keyWindow <> 0) and not
                    FindCmdLineSwitch(cmdLine, true) then
        //set existing instance
        ShowWindow(keyWindow, SW_SHOWNORMAL)
      else
        result:= true;
    end;
```

Testbarkeit Code

- Exception handling

```
{$IFDEF DEBUG}
  Application.OnException:= AppOnException;
{$ENDIF}
```

- Assert function

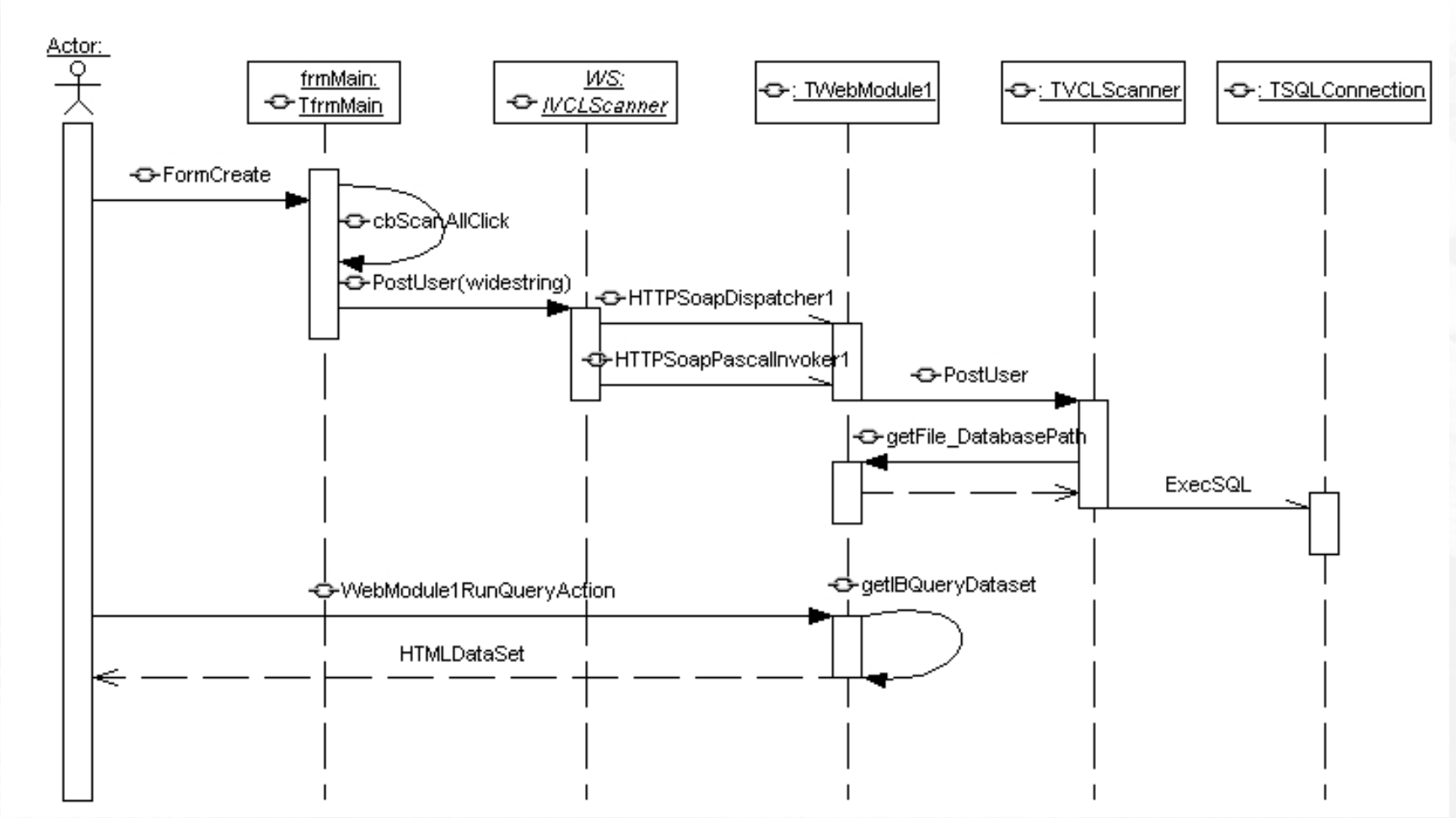
```
accObj:= TAccount.createAccount(FCustNo,
                                std_account);
assert(aTrans.checkOBJ(accObj), 'bad OBJ cond.');
```

- Logger

```
LogEvent('OnDataChange', Sender as TComponent);
LogEvent('BeforeOpen', DataSet);
```

EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Testen anhand eines SEQ



against Buffer Overflow

- The Delphi compiler hides the fact that the string variable is a heap pointer to the structure but setting the memory in advance is advisable:
- Check against Buffer overflow or TryIntToStr

```
path: string;
setLength(path, 1025);
var Source, Dest: PChar;
begin
  Source:= aSource;
  Dest:= @FData[FBufferEnd];
  if BufferWriteSize < Count then
     raise EFIFOStream.Create('Buffer over-run.');
```

Safe Heaven

- Avoid pointers as you can (unsafe code !?)
- Ex. of win32API to change:

```
pVinfo = ^TVinfo;
function TForm1.getvolInfo(const aDrive: pchar; info: pVinfo): boolean;

//refactoring from pointer to reference (const pointer with automatic deref)
function TReviews.getvolInfo(const aDrive: pchar; var info: TVinfo):
    boolean;
```

"Each pointer or reference should be checked to see if it is null. An error or an exception should occur if a parameter is invalid."

EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

less Exceptions..., more Handling

- Exception Handling und Range Checking prüfen !

```
function IsDate(source: TEdit): Boolean;
begin
  try
    StrToDate(TEdit(source).Text);
  except
    on EConvertError do
      result:= False;
  else
    result:= True;
  end;
end;
```

```
{$R+}
   SetLength(Arr,2);
   Arr[1]:= 123;
   Arr[2]:= 234;
   Arr[3]:= 345;
 {$R-}
```

Skalierbarkeit

Bei **Parametrisierung und Speicherbedarf** will man mögliche Speicherlöcher vermeiden, (try/finally Test)

```
Buffer: PChar;  //not a buffer before you use getMem
Size:= FileSize(F);
GetMem(Buffer, Size); //allocates n-Bytes on the heap
BlockRead(F, Buffer^, Size);
```

- All allocated memory needs to be freed when no longer needed. Make sure memory is released in all code paths, especially in error code paths.
- Are all objects (Database connections, Sockets, Files, etc.) freed even when an error occurs? → procexp.exe

EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Sichere Transaktion

Die Methode ApplyUpdates schreibt die Daten nicht in die DB, auch leert sie den Cache nicht. Die Daten werden zwar der DB bekanntgegeben, physikalisch muss jedoch commit folgen.

```
with query1 do begin
  database1.startTransaction;
  try
    applyUpdates;  //try to write the updates to the database
    database1.Commit;  //on succes, commit changes
  except
    database1.Rollback; //undo changes on failure
    raise: //to prevent a call to CommitUpdates!
  end;
  CommitUpdates; // on success, clear the cache
end;
```

EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07|EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

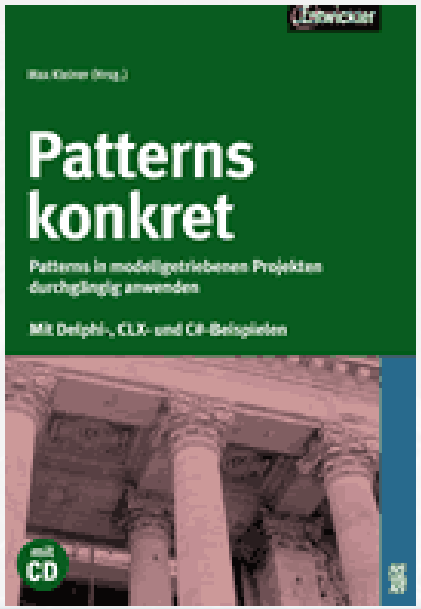
QS Review Checklist

1. Standards - are the Pascal software standards for name conventions being followed?
2. Bugs - Are the changes generally correct?
3. Are the Requirements Well Understood (Multilang)?
4. Are all program headers completed?
5. Are code changes commented appropriately?
6. Does documentation use Correct Grammar?
7. Are release notes Clear? Complete?
8. Installation Issues, Licenses, Certs. Are there any?
9. Version Control, Are output products clear?
10. Test Instructions - Are they any? Complete?

EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon

Fragen und hoffentlich Antworten ?

max@kleiner.com



EKON 11
Die Entwickler-Konferenz für Delphi, C#, JBuilder, PHP & MORE
24.-28. SEPTEMBER 2007
FRANKFURT - MÖRFELDEN
EKON 11+
07 | EuroDevCon
Teilnehmer der EKON haben freien Zugang zu allen Angeboten der EurodDevCon