

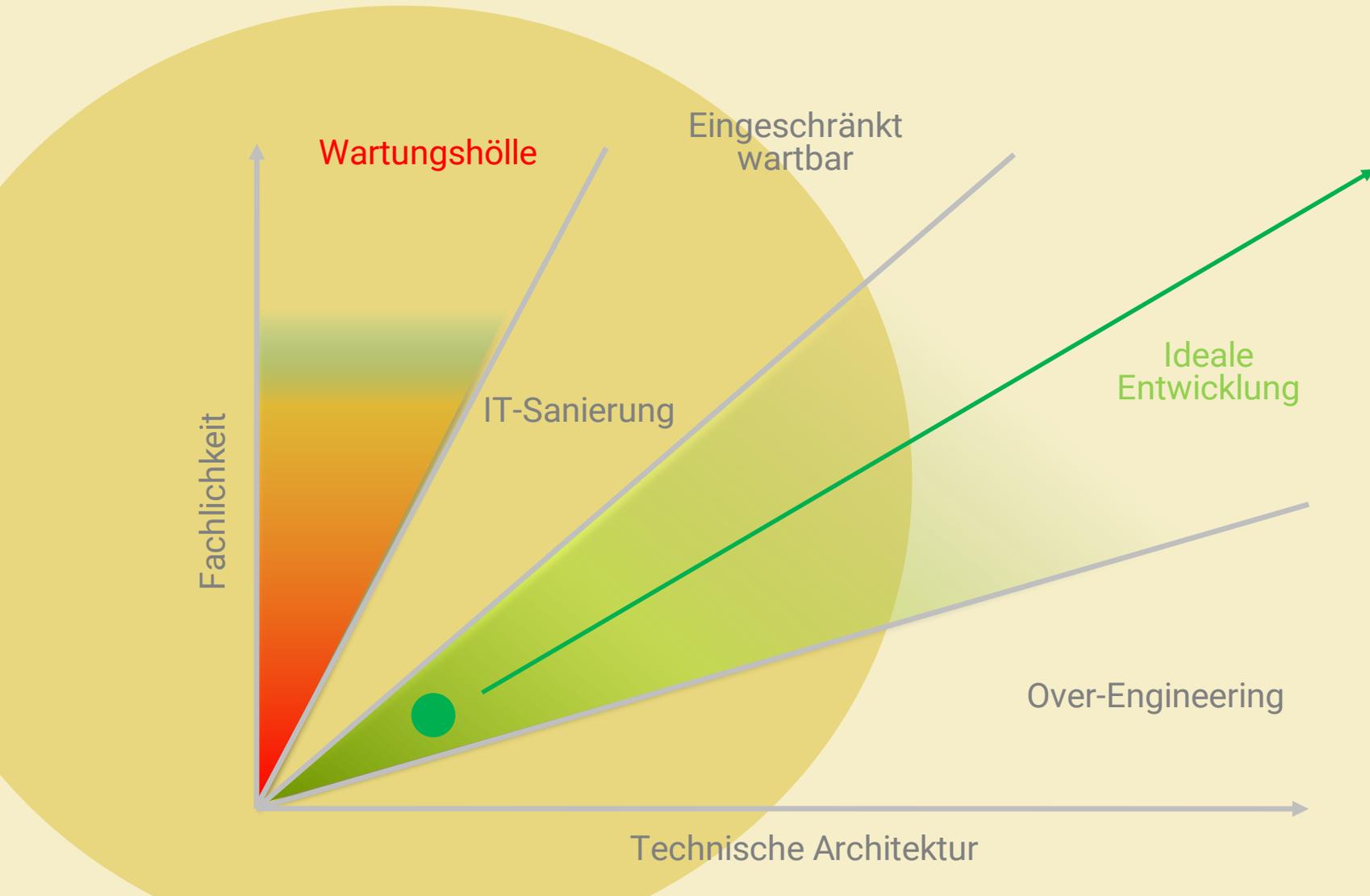
Software-Kartographie & Software-Audits

Wie erkennt Ihr gute IT-Systeme?

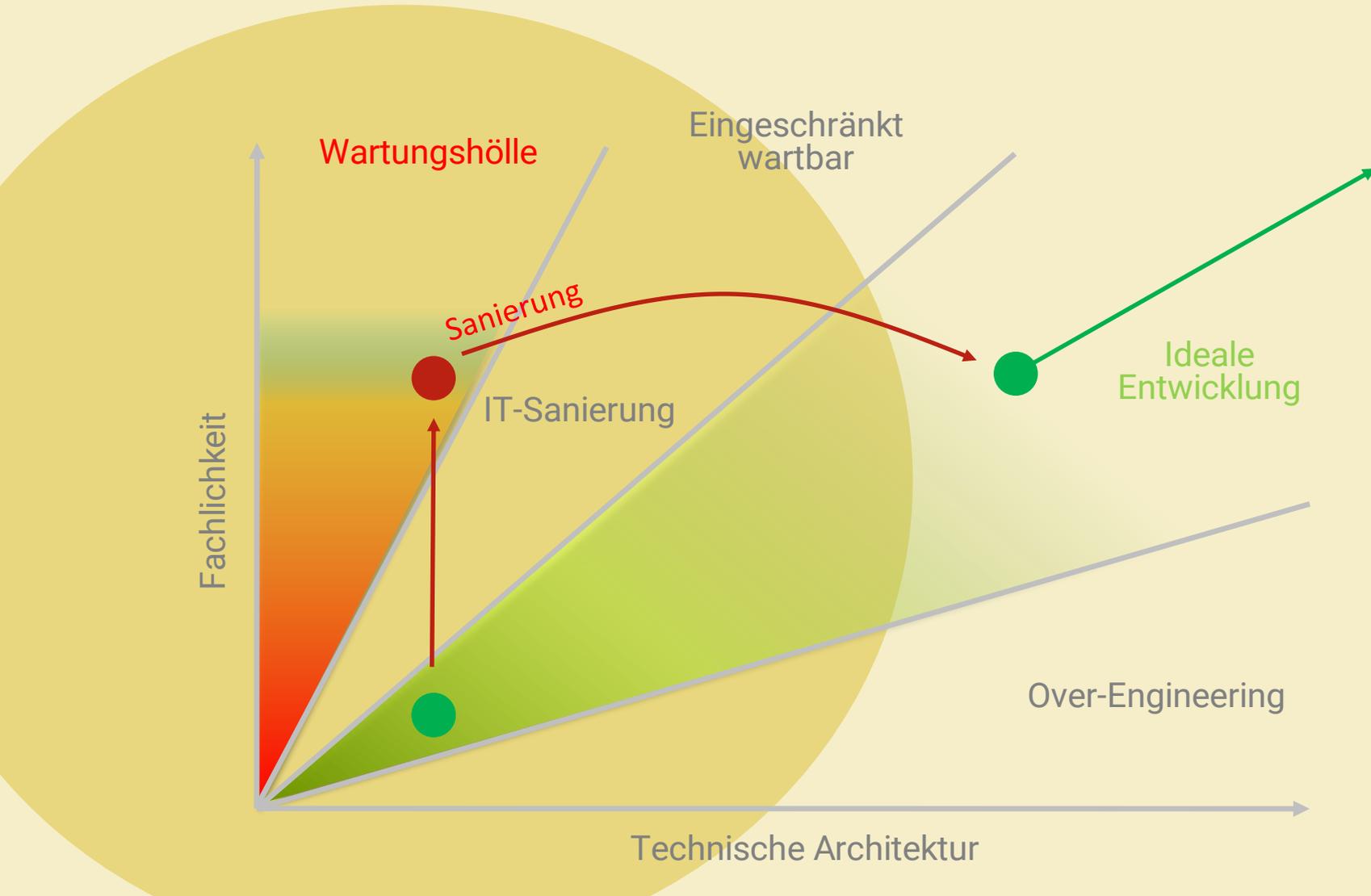
Gastvortrag
TU Darmstadt
18.01.2019

Alex Hofmann
Technischer Geschäftsführer
MaibornWolff GmbH

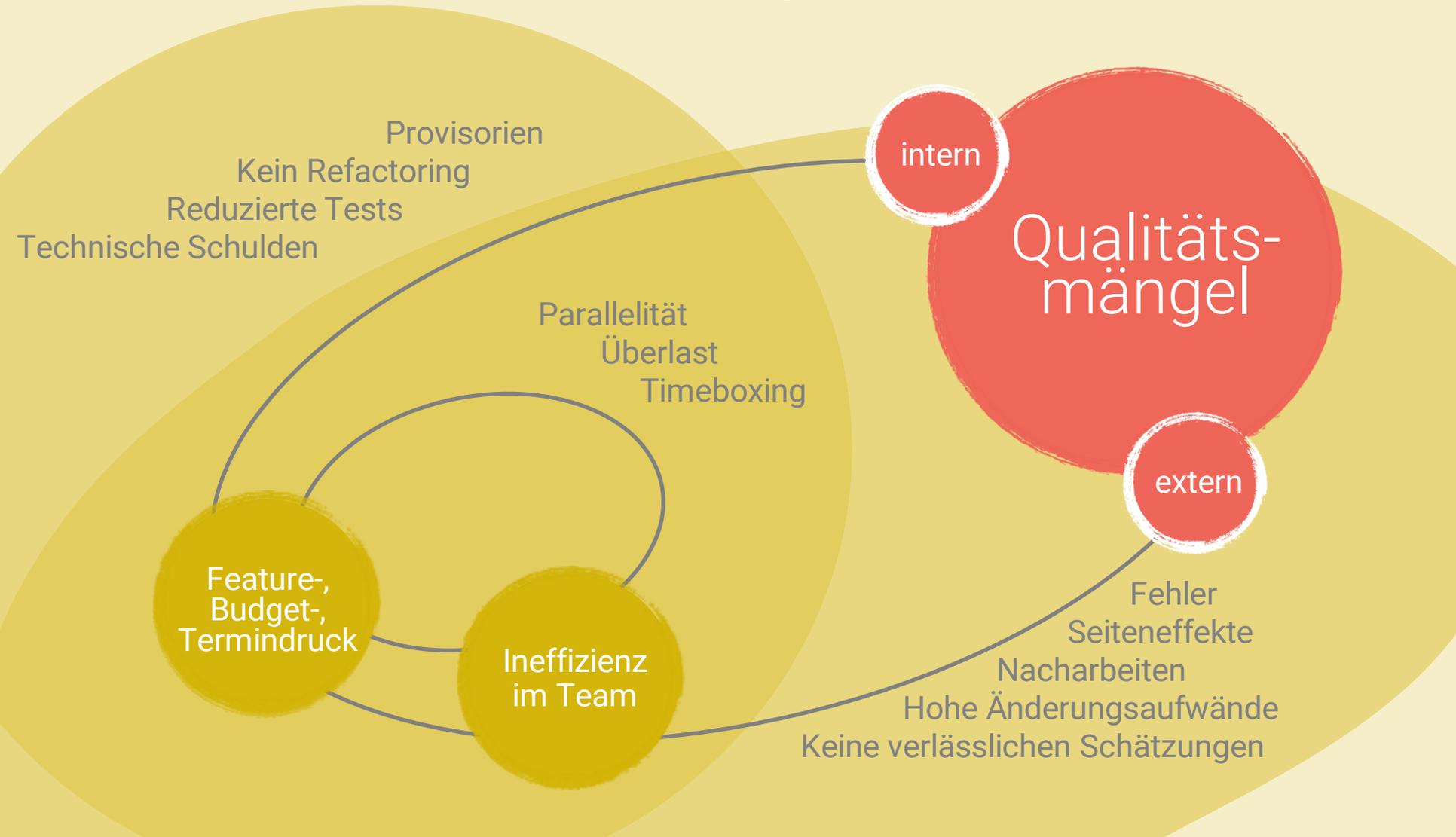
Vier Qualitäts-Korridore für Software



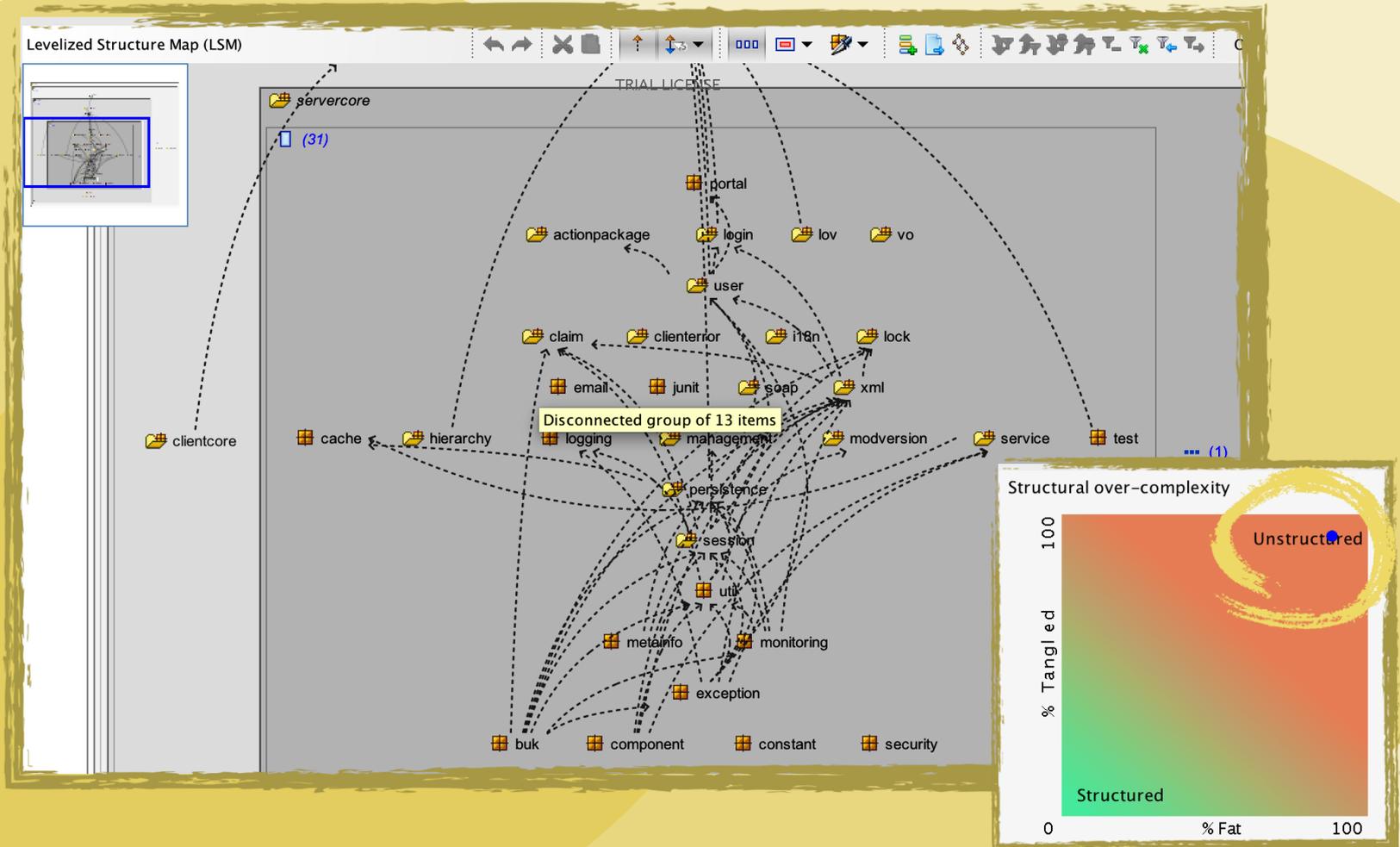
Praxis und Ideal – oft weit auseinander



Der Teufelskreis zur Wartungshölle



Hölle der Abhängigkeiten



Hölle der Verschachtelung

```
// TODO: Bisschen tief, das looping. Könnte man bestimmt schöner machen.  
Thread.Sleep(200); // milliseconds  
continue;
```

```
605 DbDataReader reader = null;  
606 try  
607 {  
608     checkCurrentTransactionIsActive(sqlStrgFunc);  
609     try  
610     {  
611         reader = cmd.ExecuteReader();  
612         _lastSqlCmdFunc = sqlStrgFunc;  
613         s_dblog.Info(m => m(MODUL + sqlStrgFunc() + " ... executed."));  
614     }  
615     catch (System.Exception e)  
616     {  
617         string sqlCmdStrg = sqlStrgFunc();  
618         s_dblog.Info(m => m(MODUL + sqlCmdStrg + " ... cancelled."));  
619  
620         // DBExceptionCheck.PrintException(e);  
621         DBExceptionCheck.CheckTransactionCompletedException(e, getTransactionNo());  
622         DBExceptionCheck.CheckOnDBServiceBusyException(e, ConnectionStrg);  
623         DBExceptionCheck.CheckOnDBServiceFailsException(e, ConnectionStrg);  
624         DBExceptionCheck.CheckTransactionDeadlockVictimException(e, sqlCmdStrg);  
625         if (DBExceptionCheck.IsCommandTimeoutExpired(e))  
626         {  
627             checkCurrentTransactionIsActive() => sqlStrgFunc()+"->Exception";  
628             // Caution: TimeoutExpired-Exception occurs after ca. 32 s  
629             // SQLCommand.CommandTimeout: 30s (default)  
630             // TransactionScope-Timeout is 60s (default)  
631             (loops < 2 && !multiSql)  
632  
633             s_log.Warn(MODUL + "SQL-Command timed out after " + loops + ". attempt (after "  
634                 + tt.ElapsedSeconds + " s) in SQL: " + sqlCmdStrg);  
635  
636             // TODO: Bisschen tief, das looping. Könnte man bestimmt schöner machen. Z.B DbConnection.TryRetry(function, 3)  
637             Thread.Sleep(200); // milliseconds  
638             continue;  
639         }  
640     }  
641 }
```

10+
Nesting Level

Hölle der Monolithen

1 Klasse, die bei jeder Systemänderung angepasst wird.

```
// maximale rekursionstiefe für zusammenbau
private const int ZUSAMMENBAUMAXDEPTH = 25;

private const int ID_TBBEMITYP = 2;

// aktuelle user session id
private int session;

#endregion

#region DataAccess for Betreuung
```

```
///
///
///
string sql =
    @"select t1.*
    FROM tk3.tk3rsnravo t1, tk3.tk3rsnr t2
    WHERE t1.ID_SACHNR = t2.ID AND t2.ID = @ID and KZ_AVOARTEN = 'P' AND t1.TBIS >= current date";

string sql =
    @"select t1.*
    FROM tk3.tk3rsnravo t1, tk3.tk3rsnr t2
    WHERE t1.ID_SACHNR = t2.ID AND t2.ID = @ID AND t1.TBIS >= current date";

var data = new ArbeitsvorgangDataSet();

var command = GetSqlCommand(sql);
ConfiguredDB.AddInParameter(command, "@ID", DbType.Int32, snrId);
InternalLoadDataSet(command, data, data.TK3RSNRAVO.TableName);
return data;
}
}
}
```

13277
Lines of Code

Den Teufelskreis durchbrechen



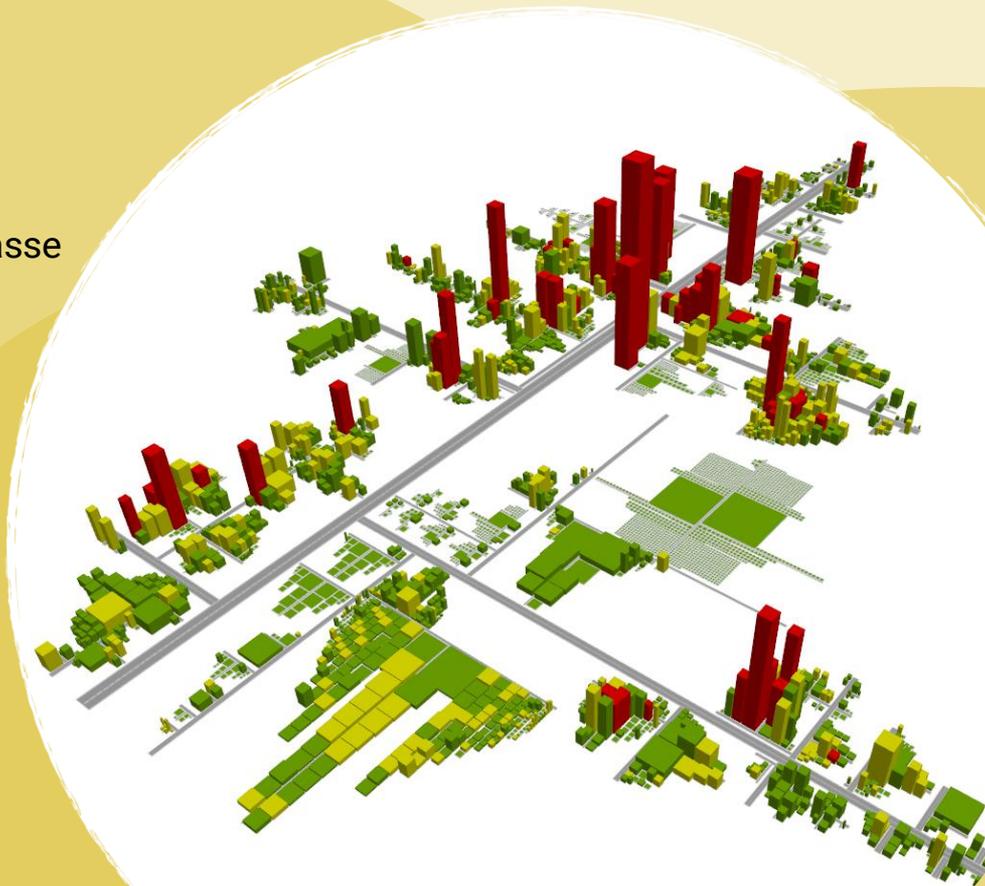
Software-
Karten

Visualisieren Sie die interne Qualität von IT-Systemen

Grundfläche:
Größe (LOC) oder
Kopplungsgrad Klasse

Höhe:
Codemetrik für
Analyse

Farbe:
Schwellwert oder
Prozessmetrik



Softwarebau
nachvollziehen

Qualität für
Weiterentwicklung
bewerten

Technische Schulden
sofort erkennen

Sie sehen Ihre Systeme wachsen

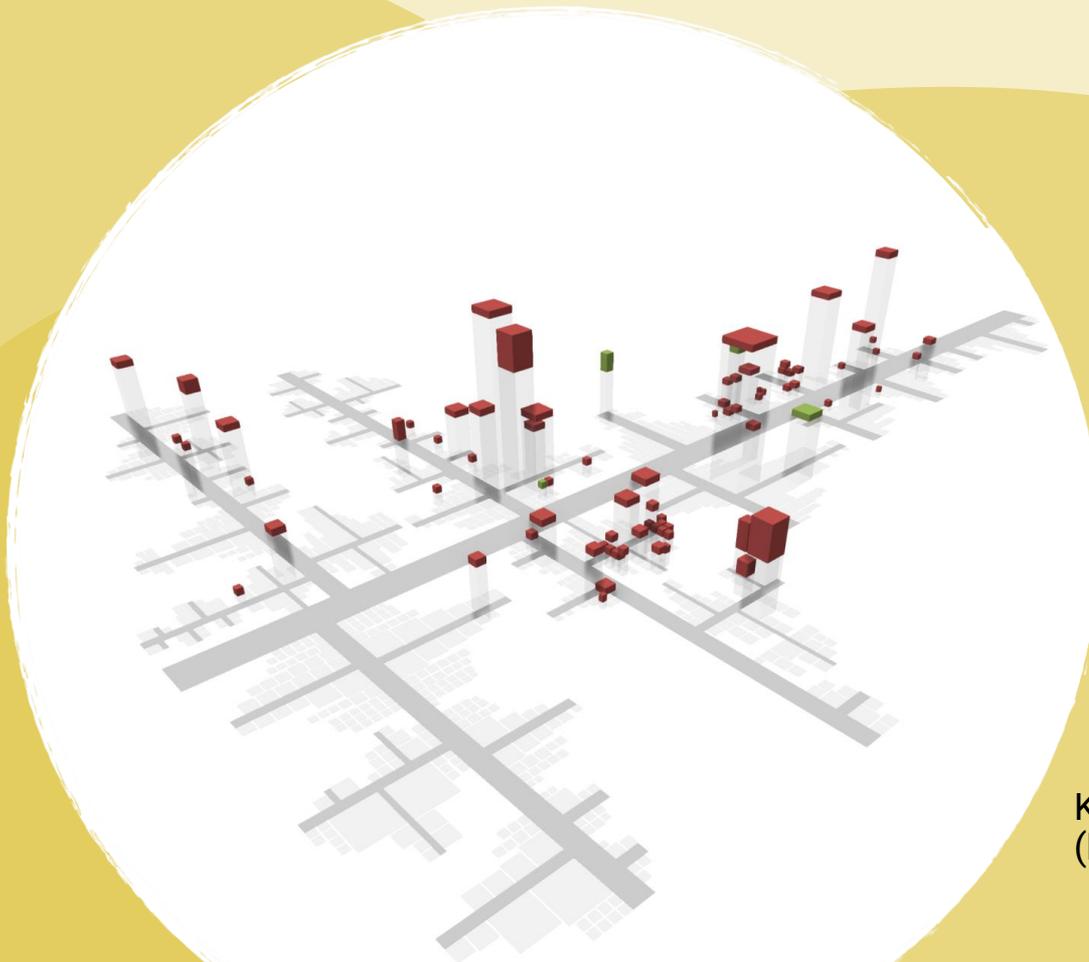
Größe
(LOC)

Kopplung
(In/Outbound)

Testabdeckung

Schachtelung
(NL+)

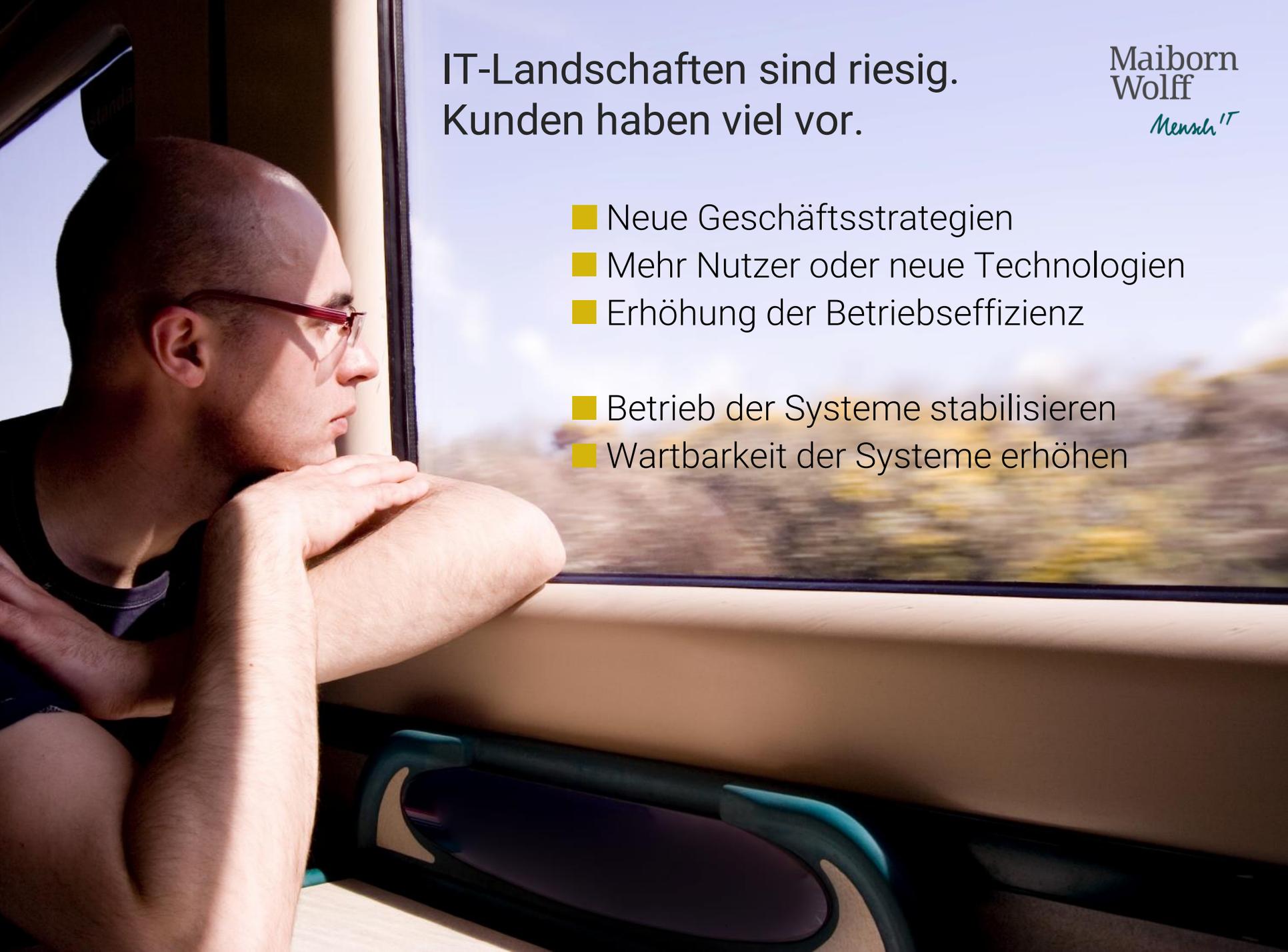
Komplexität
(McCabe)



Zwei Geschichten eines Software-Kartographen

Software-Audit

Agile Entwicklung

A man with glasses is sitting on a train, looking out the window at a blurred landscape. He is resting his chin on his hand, appearing thoughtful. The background is a bright, sunny day with a blurred view of trees and buildings.

IT-Landschaften sind riesig.
Kunden haben viel vor.

Maiborn
Wolff
Mensch IT

- Neue Geschäftsstrategien
- Mehr Nutzer oder neue Technologien
- Erhöhung der Betriebseffizienz

- Betrieb der Systeme stabilisieren
- Wartbarkeit der Systeme erhöhen

Wir auditieren 5 Sichten auf das ausgewählte IT System.

Künftige
Vorhaben

Fachliche
Architektur

Technische
Architektur/
Code

Entwick-
lungsprozess

Betrieb

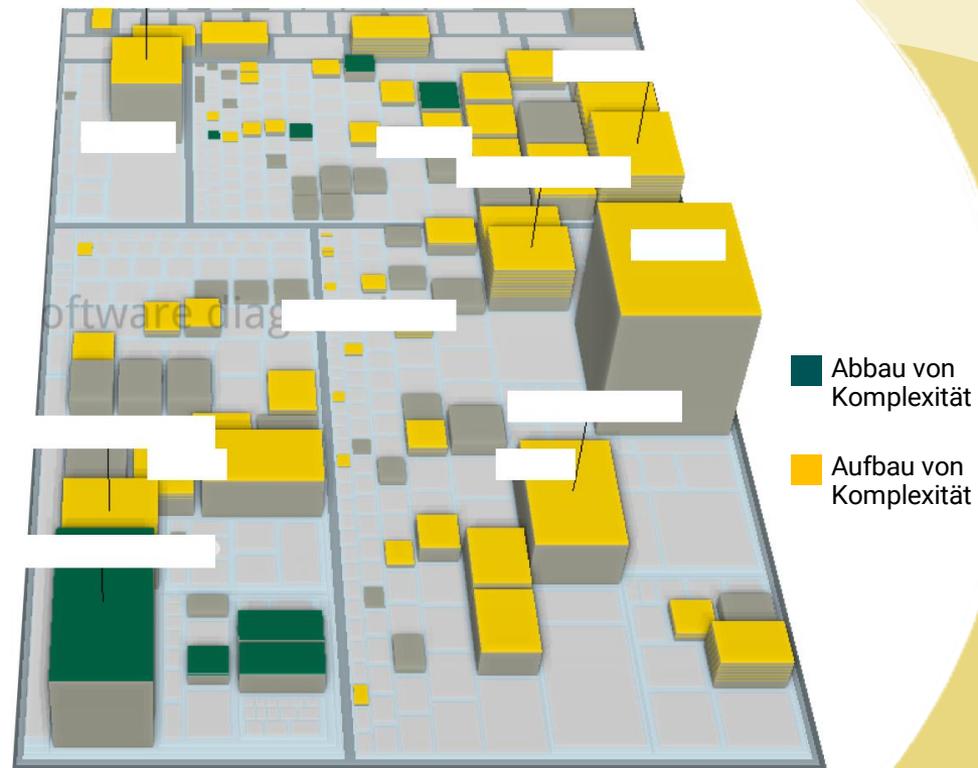


Einsatz der Softwarekarten
im Audit

Technische Schuld: Strukturen sind nicht mitgewachsen



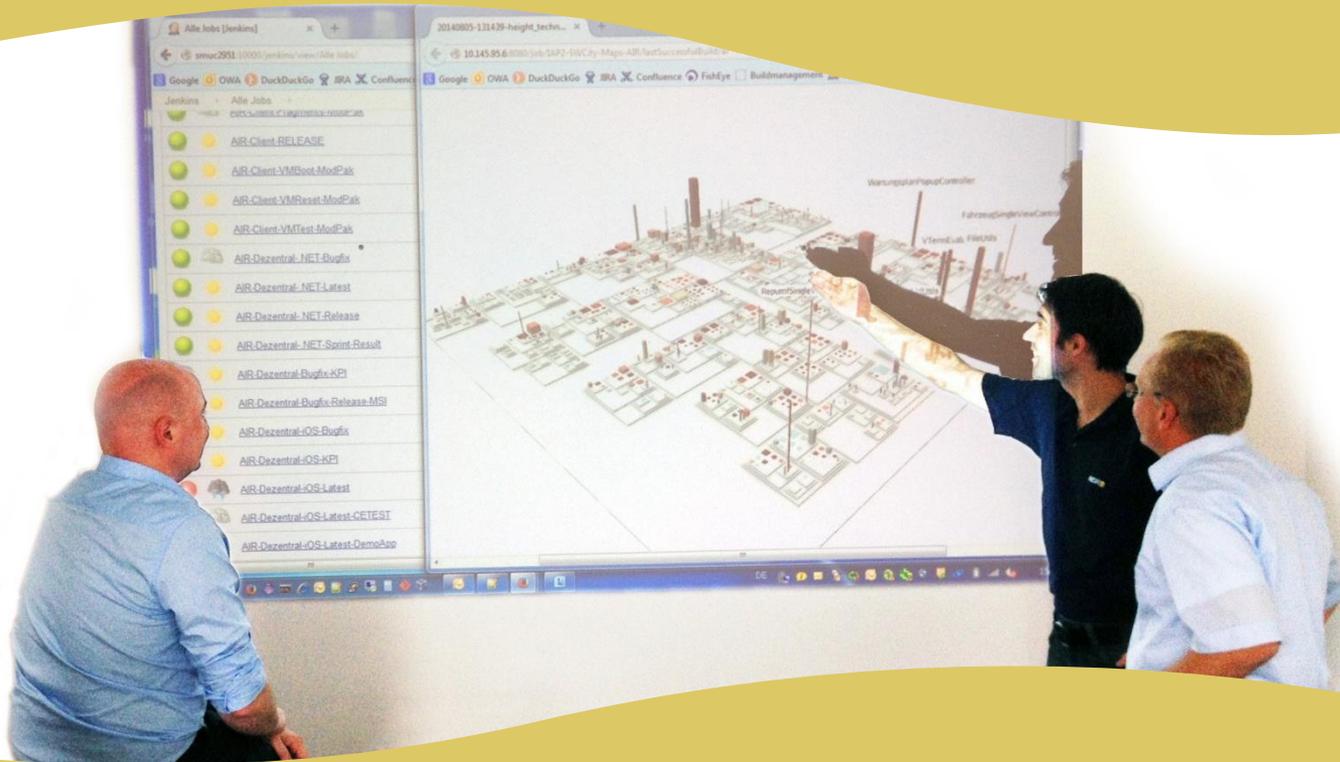
Technische Schuld: Komplexität weiter gestiegen



 seerene™

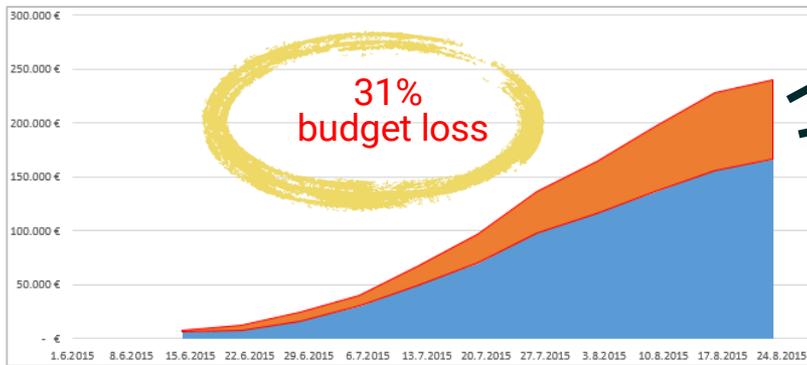
Grundfläche: Lines of Code
Höhe: McCabe-Komplexität
Farbe: Auf-/Abbau von Komplexität
Zeitraum: 03/2013 bis 03/2014 (12 Monate)

Regelmäßige Releases, regelmäßige Karten

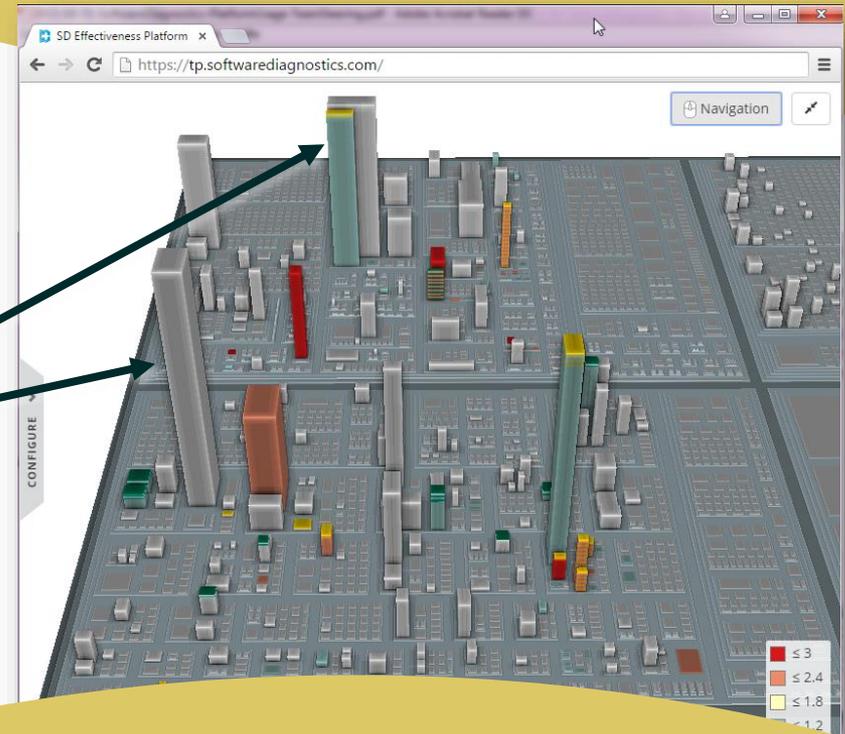


Agile Entwicklung

Blindleistung ist teuer und demotiviert



 seerene™



Agile Entwicklung

Codecharta von MaibornWolff einfach mal ausprobieren !

