

Vorteile e4

- * the new programming model
- * better control over L&F going even so far to replace widgets and even the widget technology
- * the workbench model which gives you complete control of your application

... because it is going to be to main area of innovation in the years to come.

<http://tomsondev.bestsolution.at/2012/04/13/why-is-eclipse-4-better-than-eclipse-3/>

→

Easier to add features / fix bugs

Advantage 1: Committers can fix bugs much faster

Advantage 2: Developers can add features / fix problems themselves without waiting for upstream committers to implement it for them

Advantage 3: We the platform committers don't own all the worlds problems because we have to implement it in our code and have to deal with the maintenance

lipse 3.8 will also ship as part of the Juno release, but **currently there are no 3.x releases planned after that..**

Für Migration:

the compat layer should allow you to run your applications in 4.x and together with the forward compat layer you can refactor it step by step.

I plan to write a tutorial soon about and gave a talk about it at Eclipse Con 2011 [1]. I'll hopefully be able to upload the sources to my git repo soon.

<http://eclipsesource.com/blogs/2012/06/18/migrating-from-eclipse-3-x-to-eclipse-4-e4/>

+ / -

<http://tomsondev.bestsolution.at/2012/04/13/why-is-eclipse-4-better-than-eclipse-3/>

<http://eclipse.org/e4/resources/e4-whitepaper.php>

<http://www.heise.de/developer/artikel/Eclipse-4-die-naechste-Generation-der-freien-IDE-962171.html>

Beim Thema Eventhandling hat sich bei e4 im Gegensatz zu seinem Vorgänger einiges verändert. Bei Eclipse 3.x war es Usus, eine schier unüberschaubare Anzahl an Eventhandlern zu registrieren. Mit e4 wurde ein neuer Event-Bus eingeführt, der über das typische Publish/Subscribe-Muster die Events an interessierte Handler verteilt.

Von 3 nach 4?

Viele neue Konzepte beschert e4, doch was ist mit existierenden Plug-ins und RCP-Anwendungen? Sowohl die Anzahl an Plug-ins als auch die kommerzielle Adaption von RCP sind nicht zu verachten bei einer Umstellung dieser Größenordnung. e4 bietet für den Fall eine optionale Kompatibilitätsschicht. Sie ermöglicht, existierende Plug-ins ohne Änderung auf der neuen Eclipse-Plattform benutzen zu können. Das nutzen auch die e4-Entwickler, um beispielsweise die Java Development Tools (JDT) auf e4 benutzen zu können. Eclipse 4 wird also nicht allein "die nächste Version" der Plattform, sondern parallel mit dem 3.x-Entwicklungszweig zur Verfügung gestellt. Das ermöglicht einen reibungslosen Migrationspfad ohne den großen Bruch zwischen beiden Plattformen. Zusätzlich portieren die Entwickler einige Funktionen wie die CSS Engine nach 3.x zurück.

Migration

<http://www.vogella.com/blog/2012/11/13/getting-your-eclipse-plugin-ins-ready-for-eclipse-4/>

<http://www.eclipsecon.org/2012/sessions/experiences-porting-commercial-rcp-application-eclipse-4x>

2. Problemstellung

Version 3.8 exists purely to clean up the bugs in 3.7 "Indigo", and will not be maintained after 4.3 "Kepler" is released

<http://www.eclipse.org/eclipse4/>

Mit Eclipse 4 sind einige neue Features wie das modellbasierte User Interface und auch der CSS basierte deklarative Mechanismus für das UI Styling hinzugekommen.

facilities make it easier to design and customize an application user interface.

This also adds greater flexibility to UI layout, making it easier to make a user

interface look different from an IDE

a new services-oriented programming model that makes it easier to use discreet application services
of the Eclipse platform

model-based user interface and a new CSS-based declarative mechanism for application styling

Folgende Bereiche / Faktoren dürfen nicht negativ beeinflusst werden

- Performance
- Stabilität der Anwendung
- Funktionalität der Anwendung
- Produktivität der Entwickler
- Look an Feel der Anwendung
- Usability der Anwendung

3. Lösungsansatz

< Lösungsansatz beschreiben>

4. Vorgehensmethodik

- <Gewählte Projektmethodik angeben. Beispielsweise (Wasserfall, Scrum, Hermes, RUP usw.) Dokumentation, Meilensteine, Termine, Meetings richten sich nach dieser Methode! Achtung: Alle Projektentscheide müssen am Schluss der Thesis dokumentiert und nachvollziehbar sein, unabhängig von der gewählten Methode.>

5. Aufwandsschätzung

- < Zeitaufwand der einzelnen Haupttätigkeiten aufschlüsseln>