

temperamente thesen tipps

Metriken in Projekte einführen – und überleben

Klaus Marquardt, Dräger Medical GmbH, www.kmarquardt.de

Inhalt

- temperamente
 - Veränderungsprozesse
 - Einführung von Metriken
- thesen
 - Über die Vergeblichkeit
 - Warum es trotzdem funktioniert
- tipps
 - Zielmetriken
 - Vertrauensmetriken
 - Risikometriken
- Überleben
- Erfahrungen

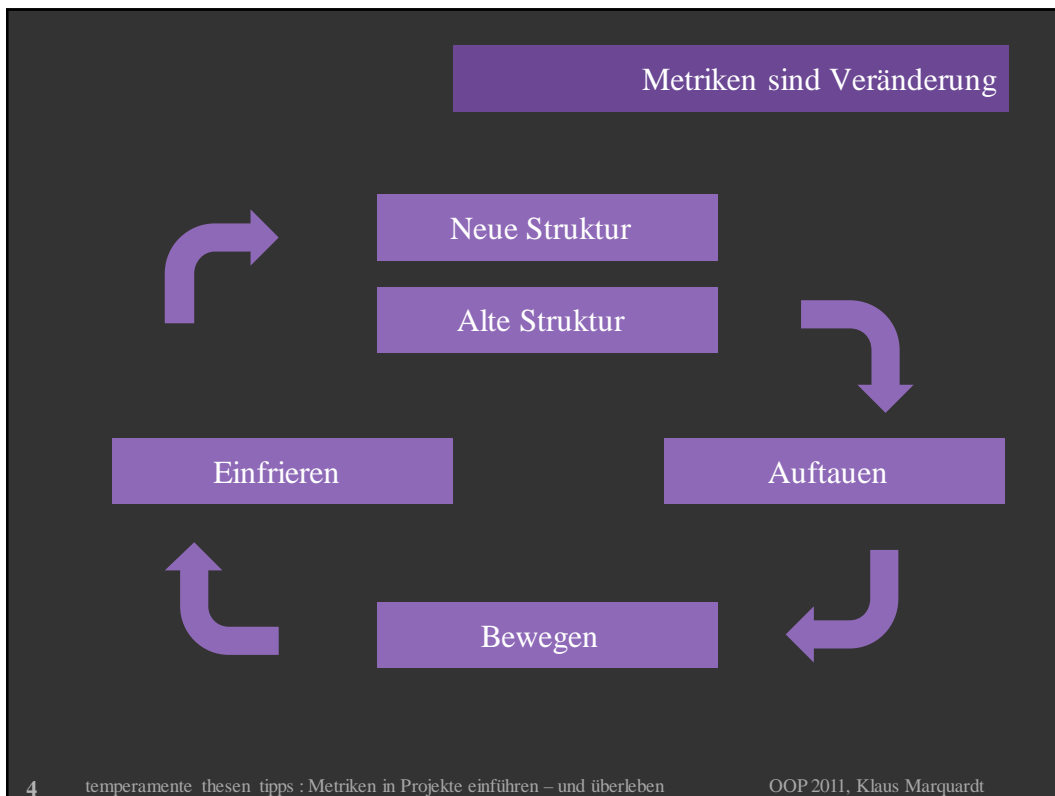
2 temperamente thesen tipps : Metriken in Projekte einführen – und überleben OOP 2011, Klaus Marquardt

Wachstum und Ergebnis				
		Eingeführte Metriken		
Fortgeführte Aktivitäten				
Auftragseingang	18.976	22.220	- 15	- 11
Umsatz	17.352	19.634	- 12	- 8
Summe Sektoren⁴				
Ergebnis Summe Sektoren	2.255	2.032	11	
in % vom Umsatz (Summe Sektoren)	13,7	11,0		
EBITDA (angepasst)	2.725	2.513	8	
in % vom Umsatz (Summe Sektoren)	16,5	13,6		
Fortgeführte Aktivitäten				
EBITDA (angepasst)	2.687	2.590	4	
Gewinn aus fortgeführten Aktivitäten	1.526	1.260	21	
Ergebnis je Aktie (in EUR) ⁵	1,70	1,43	19	
Fortgeführte und nicht fortgeführte Aktivitäten⁶				
Gewinn (nach Steuern)	1.531	1.230	24	
Ergebnis je Aktie (in EUR) ⁵	1,70	1,40	21	
Kapitalrendite (Return on Capital Employed, ROCE)				
		Q1 2010	Q1 2009	
Fortgeführte Aktivitäten				
Kapitalrendite		15,9%	12,9%	
Fortgeführte und nicht fortgeführte Aktivitäten⁶				
Kapitalrendite		16,0%	12,6%	
Free Cash Flow und Cash Conversion				
		Q1 2010	Q1 2009	
Summe Sektoren⁴				

3

temperamente thesen tips : Metriken in Projekte einführen – und überleben

OOP 2011, Klaus Marquardt

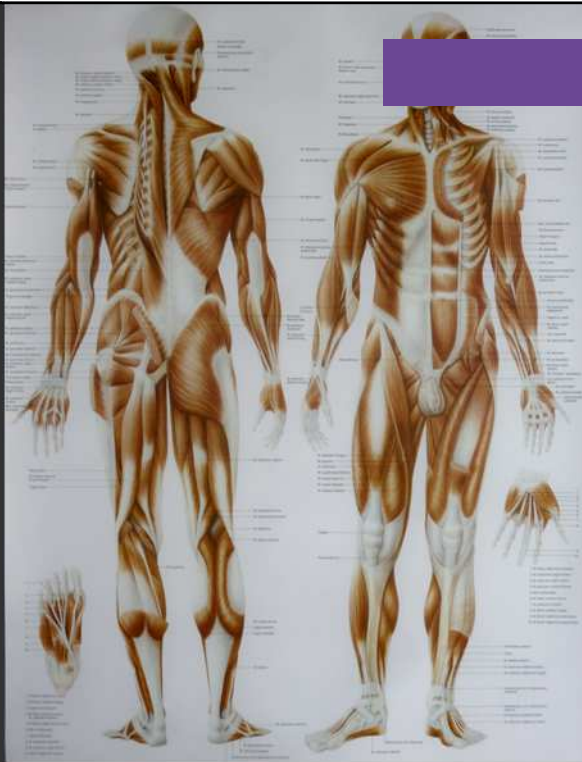


4

temperamente thesen tips : Metriken in Projekte einführen – und überleben

OOP 2011, Klaus Marquardt

Einführung von Metriken



5 temperamente thesen tips : Metriken in Projekte einführen – und überleben

OOP 2011, Klaus Marquardt

Über die Vergeblichkeit



6 temperamente thesen tips : Metriken in Projekte einführen – und überleben

OOP 2011, Klaus Marquardt

Über die Vergeblichkeit...



7

temperamente thesen tips : Metriken in Projekte einführen – und überleben

OOP 2011, Klaus Marquardt

Warum es trotzdem funktioniert



8

temperamente thesen tips : Metriken in Projekte einführen – und überleben

OOP 2011, Klaus Marquardt

Warum es trotzdem funktioniert

Bewertungszahl für die Auftretenswahrscheinlichkeit	A	Bewertungszahl für die Bedeutung	B	Bewertungszahl für die Entdeckungswahrscheinlichkeit	E
sehr hoch Sehr hohes Auftreten der Fehlerursache, unräuchbarer, ungeeigneter Prozess	9-10	sehr hoch / katastrophal Lebensgefahr, Gefahr schwerer Umweltschäden, Gefahr schwerer Verletzungen und Dauerschäden, Gefahr für Umwelt mit Folgeschäden	9-10	sehr gering Entdecken der aufgetretenen Fehlerursache ist unwahrscheinlich, die Fehlerursache wird oder kann nicht geprüft werden	9-10
hoch Fehlerursache tritt wiederholt auf, ungenauer Prozess	7-8	hoch / kritisch Gefahr von Verletzungen und Dauerschäden, Umweltbelastung, Funktionsfähigkeit des Systems oder von Teilsystemen eingeschränkt, starke Beeinträchtigung der Arbeitsbedingungen	7-8	gering Entdecken der aufgetretenen Fehlerursache ist weniger wahrscheinlich, wahrscheinlich nicht zu entdeckende Fehlerursache, unsichere Prüfungen	7-8
mäßig Gelegentlich auftretende Fehlerursache, weniger genauer Prozess	4-6	störend Erschwerung der Arbeitsbedingungen, Funktionsfähigkeitseinschränkungen von wichtigen Bedien- und Komfortsystemen	4-6	mäßig Entdecken der aufgetretenen Fehlerursache ist sehr wahrscheinlich, Prüfungen sind relativ sicher	4-6
gering Auftreten der Fehlerursache ist gering, genauer Prozess	2-3	gering Funktionseinschränkung von Bedien- und Komfortsystemen mit Arbeiterschwermis	2-3	hoch Entdecken der aufgetretenen Fehlerursache ist sehr wahrscheinlich, Prüfungen sind sicher, z. B. mehrere voneinander unabhängige Prüfungen	2-3
sehr gering Auftreten der Fehlerursache ist unwahrscheinlich	1	sehr gering Geringe Arbeiterschwermis	1	sehr hoch Aufgetretene Fehlerursache wird sicher entdeckt	1

Wie es funktioniert

- Können Sie die Veränderung auch ohne Metriken erreichen?
- Erlauben die Metriken Entscheidungen?
- Ist die Erhebung der Metriken günstig im Vergleich zu den erwogenen Maßnahmen?
- Können Sie die Erfassung der Metrik selbst vornehmen?
- Legen Sie nur den Zweck der Metrik fest:
- Geben Sie dem Team das Eigentum an der Definition der Metrik und an der Erhebung.



- Einhaltung von Terminen und Budget
- Dauer eines vollständigen Releases
- Kosten einer Installation oder eines Updates
- Anzahl der Kundenbeschwerden innerhalb der Garantie
- Anzahl und Alter der installierten Versionen

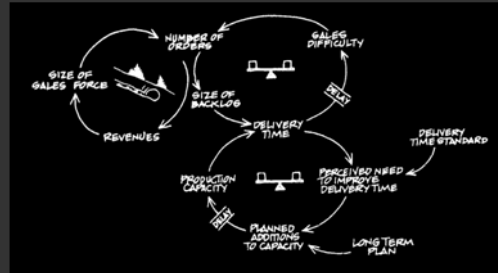
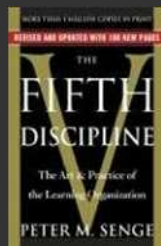
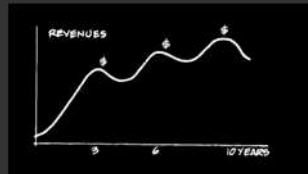
Ziele passen zum mittleren Management



Zielmetriken schaffen Vergleichbarkeit



Zielmetriken erhöhen die Komplexität



Tipps zu Zielmetriken

- Balancieren Sie kurzfristige und mittelfristige Ziele in jeder Zielvereinbarung
- Benutzen Sie Metriken, um die Qualität ihrer Entwicklungsprozesse zu messen
- Wenn Sie dringend Informationen über ein Projekt benötigen, dieses aber noch nicht beeinflussen wollen, führen Sie keine neue Metrik ein

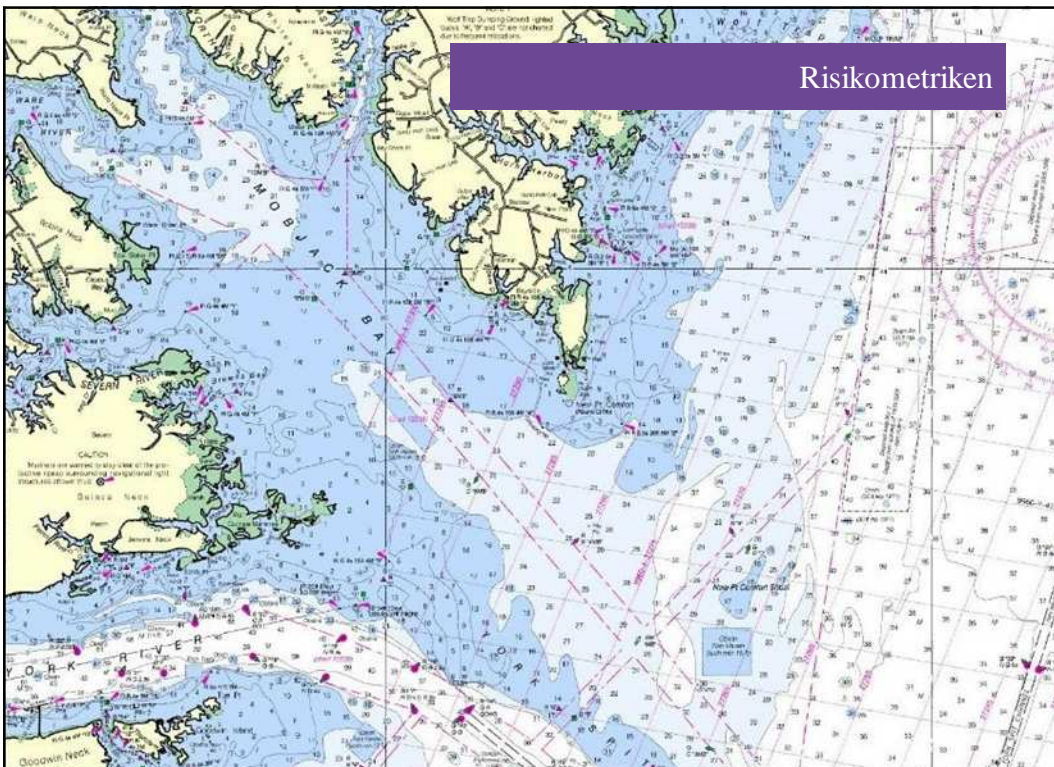


- Anzahl der fertigen Features
- Anzahl der offenen Defekte und Änderungswünsche
- Anteil des Codes, der außer Haus entwickelt wurde
- Anteil von wiederverwendeten Komponenten

Tipps zu Vertrauensmetriken

- Erheben Sie Metriken, die die Fragen eines projektfremden Beraters beantworten würden
...selbst wenn diese dem Projekt nicht unmittelbar nützen
- Bevor Sie aus Metriken Entscheidungen ableiten, fragen Sie die unmittelbar Betroffenen nach ihren Schlüssen aus den erhobenen Daten

Risikometriken



Beispiele für Risikometriken

- Fertigstellungsgrad bzw. Burn-down-Charts
- Bugfix-Rate
- Design-, Code- und Testmetriken

Tipps zu Risikometriken


- Kombinieren Sie immer Metriken miteinander, die unterschiedliche Perspektiven oder Standpunkte darstellen
- Abweichungen und Veränderungen von Metriken bieten Grund zu neuer Interpretation und zu Gegenmaßnahmen
- Ändern Sie die eingesetzten Risikometriken häufig
...oder behalten Sie diese einfach für sich

Überleben



23 temperamente thesen tips : Metriken in Projekte einführen – und überleben OOP 2011, Klaus Marquardt

Passives Überleben



24 temperamente thesen tips : Metriken in Projekte einführen – und überleben OOP 2011, Klaus Marquardt

Steuern des Überleben



25

temperamente thesen tips : Metriken in Projekte einführen – und überleben

OOP 2011, Klaus Marquardt

Mitspielen

- Motive und Ziele erkennen
 - Eigene Ziele erkennbar machen
- Klassifizierung vornehmen
 - Ziel-, Risiko- und Vertrauensmetriken
 - Vorgeschlagene Metriken verbessern
- Unschärfe darstellen
 - $\Delta x \Delta p \geq h$

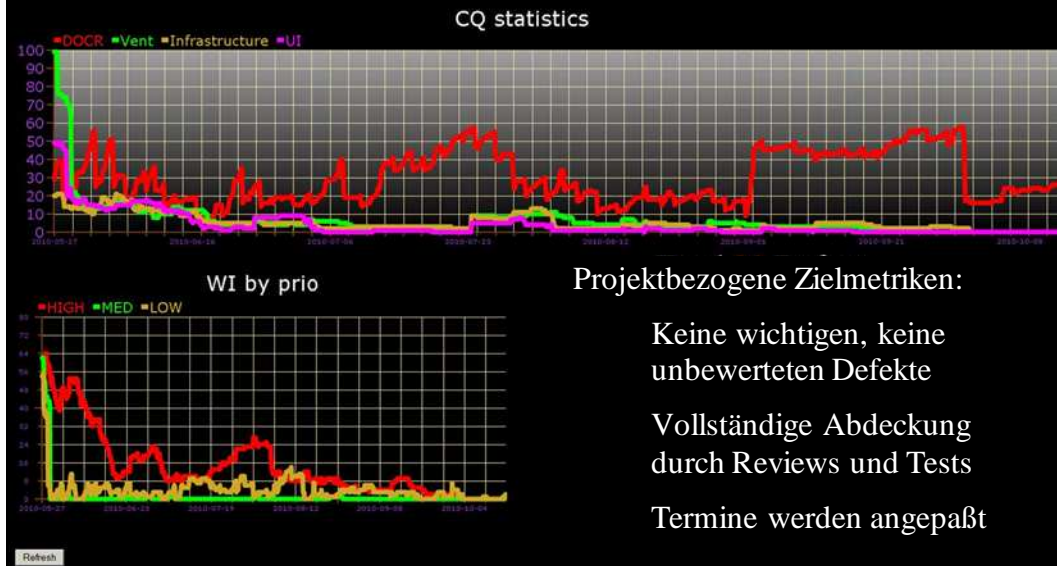


26

temperamente thesen tips : Metriken in Projekte einführen – und überleben

OOP 2011, Klaus Marquardt

Erfahrungen: Ziel-Metriken

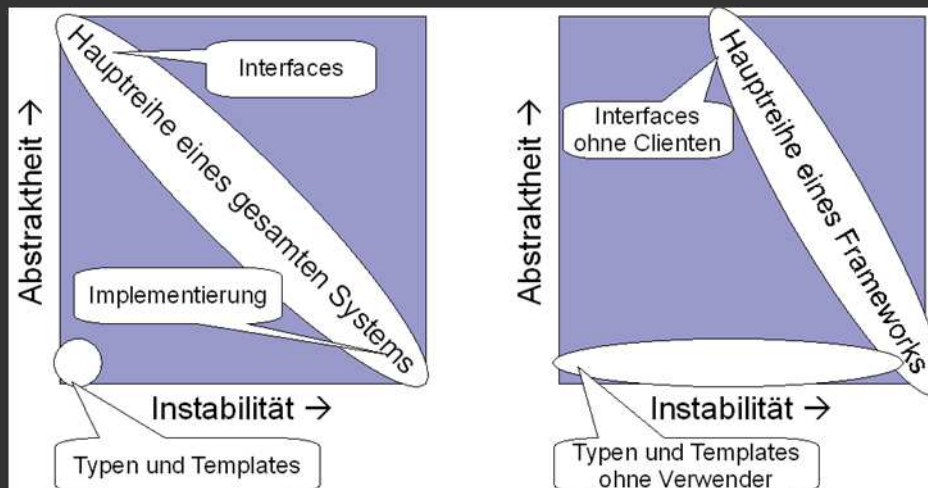


27

temperamente thesen tips : Metriken in Projekte einführen – und überleben

OOP 2011, Klaus Marquardt

Erfahrungen: Design-Metriken



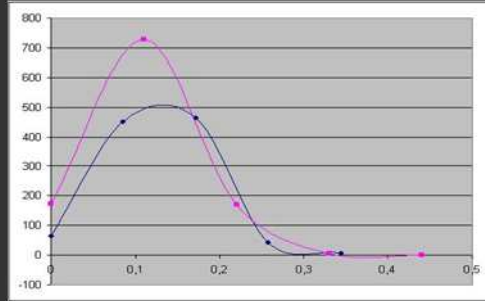
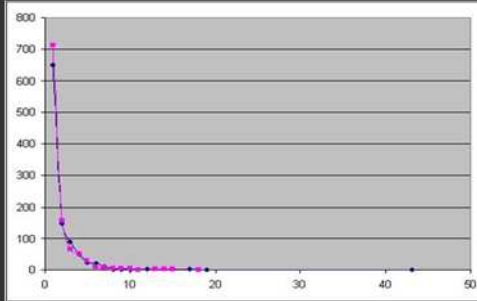
- Package-basierte Metriken
 - Robert-Martin-Metriken nur sinnvoll bei vollständigen Projekten
 - Regelmäßige Prüfung nur auf Verletzung von Abhängigkeitsregeln

28

temperamente thesen tips : Metriken in Projekte einführen – und überleben

OOP 2011, Klaus Marquardt

Erfahrungen: Code-Metriken



- Ermittlung der „suspekten Dateien“ je SW-Projekt. Kriterien:
 - Zyklomatische Komplexität einer Funktion ≥ 20
 - Parameter einer Funktion ≥ 10
 - Netto-Zeilenzahl (C++) ≥ 1000
 - „Todo“ in Kommentaren ≥ 1
 - Durchschnittliche Komplexität der Datei ≥ 0.25

29

temperamente thesen tips : Metriken in Projekte einführen – und überleben

OOP 2011, Klaus Marquardt

Code-Metriken als Risikometriken

	A	B	C	D	E	F											
1	FileName	MaxFktCom	MaxFktPara	RealLOC	ToDos	Complex / RLOC	Cyclomatic	Comments	Comment Rat	LongestLine	Functions	AvgLOCPerFunc	ParamCoun	TotalsCompl	MaxFktLines		
2	in/Des3.cpp	12	3	1420	0	0.02	23	14	0.01	108	7	202.86	14	44	59		
3	ockimpl.cpp	12	5	1269	0	0.18	232	265	0.21	189	91	13.95	78	433	85		
4	tThread.cpp	20	3	823	0	0.13	106	108	0.13	202	30	27.43	7	143	73		
5	imeUp1.cpp	21	5	787	1	0.15	120	158	0.20	212	26	30.27	17	166	120		
6	Eeprom.cpp	19	5	768	0	0.16	123	148	0.19	247	28	27.43	48	212	143		
7	Handler.cpp	60	2	735	0	0.18	135	97	0.13	150	31	23.71	17	190	369		
8	nterface.cpp	16	6	731	0	0.17	121	95	0.13	283	18	40.61	57	222	128		
9	bDialog.cpp	16	2	721	0	0.12	85	73	0.10	189	20	36.05	5	110	129		
10	ionimpl.cpp	15	4	700	0	0.15	106	140	0.20	213	49	14.29	27	208	100		
11	Vraper.cpp	20	11	695	0	0.12	81	80	0.12	394	31	22.42	46	161	185		
12	FileCtrl.cpp	20	3	670	0	0.17	116	192	0.29	403	19	35.26	20	179	175		
13	ommon.cpp	11	4	669	0	0.13	87	133	0.20	246	44	15.20	17	190	78		
14	rdimpl.cpp	8	3	662	0	0.14	93	125	0.19	191	42	15.76	41	179	68		
15	hBridge.cpp	9	4	626	0	0.17	107	94	0.15	168	41	15.27	30	194	67		
16	geimpl.cpp	24	4	600	0	0.15	89	65	0.11	180	27	22.22	19	136	207		
17	Service.cpp	11	5	592	0	0.12	70	96	0.16	178	29	20.41	19	120	97		
18	mmand.cpp	53	3	577	0	0.20	113	131	0.23	127	15	38.47	11	140	348		
19	nverter.cpp	21	8	546	0	0.13	70	47	0.09	282	17	32.12	50	144	136		
20	Module.cpp	22	2	523	0	0.17	88	68	0.13	198	19	27.53	6	113	133		
21	Dialog.cpp	10	5	521	0	0.15	76	66	0.13	133	21	24.81	37	144	83		
22	UdpPort.cpp	30	7	520	0	0.15	79	71	0.14	279	14	37.14	27	132	158		
23	Module.cpp	23	5	513	0	0.17	89	64	0.13	210	14	37.00	22	133	172		
24	Multi.cpp	16	3	497	0	0.14	76	62	0.12	177	18	28.94	28	114	65		
25	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
26	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
27	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
28	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
29	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
30	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
31	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
32	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
33	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
34	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
35	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
36	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
37	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
38	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
39	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
40	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
41	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
42	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
43	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
44	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
45	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
46	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
47	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
48	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
49	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		
50	Socket.cpp	12	2	492	0	0.17	76	62	0.12	177	18	28.94	28	114	65		

- Code-Metriken ergänzen Review-Checklisten und Lint-Output
- Dateien sind typischerweise in mehrerer Hinsicht „suspekt“
- Ausdruck einer lokalen Kultur, nur langsame Änderung
- Keine eindeutige pauschale Wertung möglich, Umgang: Warnhinweis an den Projektleiter
- Keine gesicherte Ermittlung ohne Verankerung im Release-Prozess

temperamente

Veränderungsprozesse

Einführung von Metriken

thesen

Über die Vergeblichkeit

Warum es trotzdem funktioniert

tipps

Zielmetriken

Vertrauensmetriken

Risikometriken

Überleben

Erfahrungen

Vielen Dank für Ihre Aufmerksamkeit!