

Innhold

Forord	i
1 Dødelige tegn og kommafeil	1
1.1 Kjente katastrofer	1
1.2 Feil spart er penger tjent	4
1.3 Leserveiledning	4
2 Kilder til feil	7
2.1 Dårlig problemforståelse	7
2.2 Kryptisk kode	8
2.3 Dårlig fundamentering	9
2.4 Å stole på debuggere	9
2.5 Behandle symptomene i stedet for sykdommen	10
2.6 Refaktoring	11
2.7 Oppsummering	11
3 Syv prinsipper for testing	13
3.1 Prinsipp 1. Testing kan ikke bevise at koden er feilfri	13
3.2 Prinsipp 2. Fullstendig testing er umulig	14
3.3 Prinsipp 3. Tidlig testing sparer tid og penger	15
3.4 Prinsipp 4. Feil samler seg i klynger	15
3.5 Prinsipp 5. Pass på sprøytemiddel-paradokset	16
3.6 Prinsipp 6. Testing er avhengig av konteksten	16
3.7 Prinsipp 7. Feilslutning vedrørende fravær av feil	17
3.8 Oppsummering	17
4 Arbeidsprosessen	19
4.1 Utviklingsprosessen for programvare	19
4.2 Versjonskontroll	21
4.3 Oppsummering	22

5	Testing i produktets livssyklus	25
5.1	Sentrale begreper og testaktiviteter	25
5.2	Aktivitetene i henhold til V-modellen	30
5.3	De fire testnivåene	32
5.4	De fire testtypene	38
5.5	Oppsummering	40
6	Å forhindre feil	41
6.1	Fagfellevurdering av koden	41
6.2	Statisk kildekodeanalyse	42
6.3	Tilstandskontroll	44
6.4	Versjonskontroll	44
6.5	Verktøy for å finne minnefeil	45
6.6	Enhetstester	45
6.7	Integrasjonstesting	49
6.8	Systemtester	50
6.9	Akseptansetester	50
6.10	Kontinuerlig integrasjon og automatisk testing	51
6.11	Avsluttende bemerkninger om bug-feller	51
7	Solid og testbar kode	53
7.1	Enkeltansvarsprinsippet: En klasse, ett ansvar	53
7.2	Åpen/lukket-prinsippet	56
7.3	Liskovs substitusjonsprinsipp	56
7.4	Prinsippet om separasjon av grensesnitt	57
7.5	Prinsippet om inversjon av avhengigheter	57
7.6	Avhengighetsinjeksjon	60
7.7	Veien til solid kode	60
7.8	Oppsummering	62
8	Testprosessen	63
8.1	Plan og kontroll	64
8.2	Analyse og design	65
8.3	Testimplementering og utførelse	69
8.4	Evaluerer av utgangskriterier og rapportering	71
8.5	Avsluttende testaktiviteter	72
8.6	Avsluttende ord	73
9	Statistiske testteknikker	75
9.1	Granskninger	76
9.2	Statisk analyse	77
9.3	Oppsummering	77

10	Spesifikasjonsbaserte teknikker	79
10.1	Svart boks-testing	80
10.2	Ekvivalensklasseinndeling	80
10.3	Grenseverdianalyse	83
10.4	Grenseverdianalyse og ekvivalensklasser	84
10.5	Beslutningstabelltesting	85
10.6	Tilstandsbasert testing	92
10.7	Testing basert på brukstilfeller	96
10.8	Oppsummering	100
11	Strukturbaserte testteknikker	101
11.1	Testdekning	101
11.2	Oppsummering	106
12	Erfaringsbaserte testteknikker	109
12.1	Feilgjetting	110
12.2	Utforskende testing	110
12.3	Oppsummering	113
13	Kildekodedokumentasjon	115
13.1	Ekstraherbar dokumentasjon	115
13.2	Litterær programmering	116
13.3	Hvorfor dokumentere?	117
14	Praktisk testing	119
14.1	Testrammeverk	119
14.2	Kontraktsrammeverk	120
14.3	Enhetstester i praksis	123
14.4	Eksempel på bruk av statisk assert	126
15	Testingens psykologi	129
15.1	Utvikling versus testing — to ulike tankesett	129
15.2	Uavhengig testing	130
15.3	Kunsten å formidle og motta negativ kritikk	132
15.4	Testorganisering	134
15.5	Oppsummering	135
16	Automatisert testing	137
16.1	Hva er et testverktøy?	137
16.2	Potensielle fordeler ved bruk av testverktøy	138
16.3	Risikofaktorer knyttet til bruk av testverktøy	139
16.4	Testautomatisering	140
16.5	Valg av testverktøy	144

16.6 Oppsummering	145
17 Testguide for studentprosjekter	147
17.1 Generelle råd	148
17.2 Startfasen	149
17.3 Programmets fundament – kravspesifikasjonen	151
17.4 Analyser testforholdene og design testtilfellene	152
17.5 Dynamisk testing gjennom fire testnivåer	153
17.6 Regresjonstesting	155
17.7 En siste test	155
17.8 Testing kan ikke bevise at koden er feilfri	156
17.9 Avsluttende ord	156
A En fikspunktimplementering	157
A.1 Flyttall og fikspunkttall	157
A.2 Implementering av fikspunktklasse	159
A.3 Enhetstester	174
A.4 Testdekning for FixedPoint-klassen	186
A.5 Matematiske funksjoner	197
A.6 Testdekning for matematiske funksjoner	217
A.7 Kildekodedokumentasjon	226
Norsk–engelsk ordliste	227
Engelsk–norsk ordliste	231
Bibliografi	235
Stikkordregister	239