

# Spis treści

<b>1. Wstęp do tomu trzeciego</b> ( <i>Jan Werewka</i> ) .....	7
1.1. Klasyczne i zwinne metodyki zarządzania projektami .....	7
1.2. Zakres opracowania .....	9
Literatura .....	10
<b>2. Klasyczne zarządzanie projektami informatycznymi na przykładzie PMBOK</b> ( <i>Jan Werewka</i> ) .....	11
2.1. Wstęp .....	11
2.2. Cele PMI .....	11
2.3. PMBOK – rozwój i zasady .....	15
2.4. Zdarzenia – grupowanie procesów według cyklu projektowego.....	16
2.5. Artefakty .....	20
2.6. Role .....	28
2.7. Procesy .....	35
2.8. Dyscypliny PMBOK .....	43
2.9. Narzędzia i techniki metodyki PMBOK .....	45
2.10. Wnioski .....	51
Literatura .....	51
<b>3. Systematyczny opis metodyki Scrum dla zespołów projektowych</b> ( <i>Jan Werewka, Michał Turek, Tomasz Włodarek</i> ) .....	53
3.1. Zagadnienie konstrukcji systematycznego opisu metodyk wytwarzania i zarządzania .....	53
3.2. Scrum – geneza i rozwój .....	55
3.3. Postulaty metodyk zwinnych .....	57
3.3.1. Scrum jako proces empiryczny – filary procesu .....	60
3.3.2. Scrum jako metodyka normatywna (szkielet pracy) .....	60
3.4. Zdarzenia metodyki Scrum .....	61
3.5. Artefakty .....	65

3.6. Role .....	67
3.7. Procesy .....	71
3.7.1. Planowanie sprintu .....	71
3.7.2. Rozwój produktu (część wykonawcza sprintu) .....	73
3.7.3. Kontrola i monitorowanie sprintu .....	74
3.7.4. Przegląd sprintu .....	76
3.7.5. Retrospektywa sprintu .....	77
3.7.6. Realizacja rekomendacji retrospektywy .....	78
3.8. Zestawienie składowych i pojęć metodyki Scrum .....	79
3.9. Podsumowanie .....	81
Literatura .....	82
<b>4. Zarządzanie projektem w przedsiębiorstwie według metodyki Scrum</b> ( <i>Jan Werewka, Michał Turek</i> ) .....	83
4.1. Wstęp .....	83
4.2. Zdarzenia metodyki Scrum w zarządzaniu projektem .....	83
4.3. Zestawienie procesów metodyki Scrum w zarządzaniu projektem .....	84
4.3.1. Opracowanie wizji projektu .....	86
4.3.2. Rozpoczęcie projektu .....	87
4.3.3. Planowanie wydania .....	88
4.3.4. Wydanie produktu .....	90
4.3.5. Zamknięcie projektu .....	90
4.3.6. Monitorowanie wydania .....	90
4.3.7. Gromadzenie wymagań .....	91
4.3.8. Pielęgnacja (utrzymanie) rejestru produktowego .....	92
4.3.9. Usuwanie przeszkód .....	93
4.4. Warsztat .....	93
4.4.1. Praktyki .....	93
4.4.2. Narzędzia .....	94
4.4.3. Techniki .....	98
4.5. Skalowanie Scrum .....	102
4.6. Podsumowanie .....	104
Literatura .....	105
<b>5. Programowanie ekstremalne (XP)</b> ( <i>Michał Turek</i> ) .....	106
5.1. Wprowadzenie .....	106
5.2. XP a Scrum .....	107
5.3. Uniwersalne zasady (pryncypia) .....	107
5.4. Artefakty .....	109
5.5. Role .....	112
5.6. Procesy .....	113

5.7. XP – wytwarzanie kodu .....	116
5.7.1. Programowanie w parach .....	116
5.7.2. Standaryzacja kodu źródłowego .....	116
5.7.3. Ciągła integracja .....	117
5.7.4. Współwłasność kodu .....	117
5.7.5. Refaktoryzacja .....	117
5.7.6. 40-godzinny tydzień pracy .....	118
5.7.7. Współpraca z klientem .....	118
5.8. Zarządzanie ryzykiem w projektach XP .....	119
5.9. Podsumowanie .....	121
Literatura .....	121
<b>6. Zarządzanie jakością oprogramowania: wprowadzenie do testowania</b>	
<i>(Grzegorz J. Nalepa, Krzysztof Kaczor)</i> .....	123
6.1. Podstawy testowania .....	123
6.1.1. Wstęp .....	123
6.1.2. Zasady testowania .....	124
6.1.3. Proces testowy .....	125
6.1.4. Psychologia testowania .....	126
6.2. Testowanie w procesie wytwarzania oprogramowania .....	127
6.2.1. Poziomy testów .....	128
6.2.2. Typy i cele testów .....	130
6.2.3. Testowanie w fazie utrzymania .....	131
6.3. Testowanie statyczne .....	132
6.4. Projektowanie testów .....	135
6.4.1. Techniki czarnoskrzynkowe .....	135
6.4.2. Techniki białoskrzynkowe .....	136
6.4.3. Techniki oparte na doświadczeniu .....	137
6.5. Zarządzanie procesem testowym .....	138
6.5.1. Niezależność testowania .....	138
6.5.2. Planowanie testowania .....	138
6.5.3. Nadzór testowania .....	141
6.5.4. Zarządzanie incydentami .....	141
6.6. Klasy narzędzi testowych .....	142
6.6.1. Narzędziowe wsparcie testowania statycznego .....	142
6.6.2. Narzędziowe wsparcie tworzenia specyfikacji testów .....	143
6.6.3. Narzędziowe wsparcie wykonywania testów .....	143
6.6.4. Narzędziowe wsparcie testowania wydajności .....	144
6.6.5. Narzędzia wspierające testowanie w określonych dziedzinach .....	144
6.6.6. Wybrane otwarte narzędzia wspomagające testowanie .....	144
6.7. Podsumowanie .....	146
Literatura .....	146

<b>7. Środowiska wytwarzania oprogramowania: wybrane narzędzia</b>	
(Grzegorz J. Nalepa, Szymon Bobek) .....	147
7.1. Wstęp .....	147
7.2. Zintegrowane środowiska programistyczne .....	147
7.2.1. NetBeans .....	147
7.2.2. Eclipse .....	148
7.2.3. KDevelop .....	149
7.3. Systemy wersjonowania i zarządzania zmianami kodu .....	150
7.3.1. RCS – protoplasta systemów wersjonowania.....	150
7.3.2. Scentralizowane systemy wersjonowania.....	151
7.3.3. Podejście rozproszone do wersjonowania .....	152
7.3.4. Systemy wersjonowania dostępne <i>on-line</i> .....	154
7.4. Systemy wspomagające dokumentowanie kodu.....	155
7.4.1. JavaDoc .....	156
7.4.2. Doxygen.....	158
7.5. Systemy kontroli i śledzenia błędów .....	161
7.6. Podsumowanie .....	163
Literatura.....	163