

# Obsah

Předmluva	9
Úvod	10
<b>1.část    Problém</b>	<b>15</b>
<b>1. Základní problém: riziko</b>	<b>16</b>
<i>Vývoj programového vybavení selhává a nedodává požadované hodnoty. Toto selhávání má obrovský vliv na ekonomický a lidský potenciál. Musíme najít nový způsob, jak software vyvíjet.</i>	
<b>2. Vývojová epizoda</b>	<b>19</b>
<i>Každodenní programování postupuje od úkolu, který je jasně spojen s funkcí požadovanou zákazníkem, přes testy, implementaci a návrh až po integraci. Malá část všech činností vývoje softwaru je vložena do všech epizod.</i>	
<b>3. Ekonomická stránka vývoje softwaru</b>	<b>21</b>
<i>Ekonomickou hodnotu našeho vývoje softwaru musíme zvýšit tím, že budeme vydávat peníze pomaleji a příjmy získávat rychleji. Je pravděpodobné, že tím zvýšíme délku produktivního života projektu. Ale nejvíc ze všeho musíme rozšířit možnosti výběru pro naše rozhodování.</i>	
<b>4. Čtyři proměnné</b>	<b>24</b>
<i>V našich projektech budeme ovládat čtyři proměnné: náklady, čas, kvalitu a šíři zadání. Nejcennější formu řízení nám poskytuje jedna z nich: širší zadání.</i>	
<b>5. Náklady na změnu</b>	<b>28</b>
<i>V určitých okolnostech lze exponenciální růst nákladů na software měnící se v čase vyrovnat. Pokud můžeme křivku vyrovnat, přestanou platit staré předpoklady o nejlepší způsobu, jak vyvíjet software.</i>	
<b>6. Učíme se řídit</b>	<b>32</b>
<i>Vývoj programového vybavení musíme řídit prováděním mnoha malých korekcí – nikoli několika velkými úpravami – podobně jako při řízení auta. To znamená, že budeme potřebovat zpětnou vazbu abychom věděli, kdy jsme trochu odbočili. Budeme potřebovat mnoho příležitostí na provádění korekcí a musíme být schopni tyto korekce dělat za přijatelné náklady.</i>	

<b>7. Čtyři hodnoty</b>	<b>34</b>
<i>Budeme úspěšní, když budeme mít styl zachovávající důslednou sadu hodnot, které slouží potřebám člověka i společnosti: komunikace, jednoduchost, zpětná vazba a odvaha.</i>	
<b>8. Základní principy</b>	<b>39</b>
<i>Ze čtyř hodnot odvodíme zhruba tucet základních principů, které povedou náš nový styl. Navrhované vývojové postupy prověříme abychom zjistili, jak se s těmito principy poměří.</i>	
<b>9. Zpět k základům</b>	<b>45</b>
<i>Chceme udělat vše co musíme, abychom docílili stabilního a předvidatelného vývoje softwaru. Ale na cokoli navíc nemáme čas. Čtyřmi základními vývojovými činnostmi jsou psaní zdrojového textu, testování, poslouchání a navrhování.</i>	
<b>2. část    Řešení</b>	<b>51</b>
<b>10. Zběžný přehled</b>	<b>52</b>
<i>Spolehne se na synergie mezi jednoduchými postupy, které byly před mnoha lety často opouštěny pro svou údajnou nepraktičnost nebo naivitu.</i>	
<b>11. Jak by to mělo fungovat?</b>	<b>59</b>
<i>Postupy se vzájemně podporují. Nedostatek jednoho postupu pokryjí přednosti těch ostatních.</i>	
<b>12. Strategie řízení</b>	<b>65</b>
<i>Celkový projekt budeme řídit pomocí základních pravidel obchodu: vývoj po fázích, rychlá a konkrétní zpětná vazba, jasná artikulace potřeb na vyvíjený systém a využití odborníků na speciální úkoly.</i>	
<b>13. Strategie vybavení</b>	<b>69</b>
<i>Pro náš tým vytvoříme otevřené pracoviště s malými privátními prostory na jeho obvodu a společnou programovací oblastí uprostřed.</i>	
<b>14. Rozdělení odpovědností mezi vývoj a vedení společnosti</b>	<b>72</b>
<i>Jedním klíčem naší strategie je udržovat soustředění technických lidí na technické problémy a vedoucích pracovníků na problémy společnosti. Projekt musí být řízen rozhodováním vedení, ale tato rozhodnutí musí obsahovat technická rozhodnutí o nákladech a riziku.</i>	

<b>15. Strategie plánování</b>	<b>75</b>
<i>Plánujeme tak, že rychle načrtneme celkový plán, který budeme upřesňovat více a více v čím dál tím kratších časových intervalech: roky, měsíce, týdny, dny. Plán vytvoříme rychle a levně, takže bude mít malou setrvačnost, když jej musíme měnit.</i>	
<b>16. Strategie vývoje</b>	<b>84</b>
<i>Na rozdíl od strategie řízení představuje strategie vývoje radikální odklon od zdravého selského rozumu. Budeme pečlivě vytvářet řešení dnešního problému dnes a věřit tomu, že zítřejší problém vyřešíme zítra.</i>	
<b>17. Strategie návrhu</b>	<b>88</b>
<i>Budeme nepřetržitě utvářet návrh systému a začneme od samotného jednoduchého počátku. Zbavíme se veškeré flexibility, která se neosvědčí.</i>	
<b>18. Strategie testování</b>	<b>96</b>
<i>Napišeme testy vždy o něco dřív, než vlastní zdrojový text. Tyto testy zachováme jednou provždy a budeme je často spouštět všechny najednou. Rovněž tak si odvodíme testy z pohledu zákazníka.</i>	
<b>3. část Implementace XP</b>	<b>99</b>
<b>19. Přijetí XP</b>	<b>100</b>
<i>XP budeme přijímat vždy jen po jednom postupu, který pokryje ten nejpalcivější problém našeho týmu. Jakmile se již tento problém nejeví akutní, přejdeme k dalšímu.</i>	
<b>20. Postupné zavádění XP</b>	<b>101</b>
<i>Projekty, které chtějí změnit svou současnou kulturu, jsou daleko běžnější než projekty, které mohou vytvořit novou kulturu od základu. U probíhajících projektů budeme XP zavádět postupně po menších částech: začneme testováním nebo plánováním.</i>	
<b>21. Životní cyklus ideálního projektu XP</b>	<b>105</b>
<i>Ideální projekt XP projde krátkou úvodní vývojovou fází, kterou následují léta paralelní provozní podpory a zdokonalování, a nakonec elegantní odchod do výslužby, když už projekt nemá žádný smysl.</i>	

<b>22. Lidské role</b>	<b>110</b>
<i>Aby mohl extrémní tým fungovat, musí se naplnit určité role: programátor, zákazník, kouč a stopař.</i>	
<b>23. Pravidlo 20-80</b>	<b>117</b>
<i>Úplná hodnota XP se projeví až poté, co budou zavedeny všechny postupy. Mnoho postupů lze přijmout jeden po druhém, ale jejich účinky se znásobí, jakmile budou používány současně.</i>	
<b>24. Co ztěžuje XP</b>	<b>118</b>
<i>Ačkoli mohou programátoři vykonávat jednotlivé postupy, je těžké složit všechny součásti a udržet je pohromadě. Co ztěžuje XP jsou především emoce a zejména strach.</i>	
<b>25. Kdy bychom neměli XP zkoušet</b>	<b>121</b>
<i>Přesné hranice XP ještě nejsou zcela jasné. Některé absolutní překážky, které zabrání XP v činnosti, však existují: velké týmy, nedůvěřiví zákazníci a technologie, které nepodporují elegantní změny.</i>	
<b>26. XP v činnosti</b>	<b>124</b>
<i>XP se umí přizpůsobit běžným formám smluv, i když s drobnými úpravami. Zejména smlouvy typu pevná cena za pevnou šíři zadání se stanou smlouvami typu pevná cena za pevný termín a zhruba pevnou šíři zadání, jestliže budou probíhat současně s „Plánovací hrou“.</i>	
<b>27. Závěr</b>	<b>128</b>
<b>Bibliografie opatřená poznámkami</b>	<b>130</b>
<b>Slovníček</b>	<b>137</b>
<b>Rejstřík</b>	<b>140</b>