Appendix 2
Anarchy Questionnaire

The following questions reference something called "Good Stuff". "Good Stuff" for the purposes of this questionnaire means those parts of the architectural concepts, checklists and vocabulary of the first 4 chapters that are to be introduced into "the company." The phrase "the company" means the people in the group, division, department, team etc. who are evolving into developers, designers, managers, stakeholders who will be shifting to and using or are otherwise impacted by Good Stuff.

Each item below is about a topic in software engineering or an aspect of shifting to Good Stuff. Each item has a scale from 1 to 10. The sentence at the left of each item is the positive extreme of cooperation with or interest in the shift to Good Stuff. The leftmost sentence may also be thought of as the most disciplined aspect of that item's topic. The sentence at the right hand side is a worst-case statement of an aspect of anarchy or resistance or impediment to the shift. A "1" then, indicates that the item is not a problem or risk. A "10" indicates a significant or potentially significant problem or risk. Any given issue may, of course, be somewhere in between.

For each item, place a circle around the one number which represents your assessment of where the company (department, team etc.) best fits, either between or at the extreme. When answering assume the perspective of a "evolvee," that is, someone who will be required to learn about Good Stuff or who will be affected by the adoption of Good Stuff.

We invite you to feel free to be honest. Specifically, it will not be helpful to downplay real problems. The more accurately you describe your team and its environment now, the easier it will be later to correctly assess needed resources.

When you have finished answering all the questions, add up all the values and divide by 20 to get an average score.

Please consult Chapter 5 of the book Timing is Almost Everything for interpretation of the scores.

1. You believe that there will be rewards from the company in the form of something you truly value if you help accomplish making the shift towards Good Stuff.					You believe there will be no special rewards.				
1	2	3	4	5	6	7	8	9	10

2. Shifting to Good Stuff represents a good fit with the company's past values.					Good Stuff represents certain values that conflict with what most people at the company really believe and how they actually behave.				
1	2	3	4	5	6	7	8	9	10

3. You perceive strong political support for a shift to Good Stuff.					You perceive weak political support.				
1	2	3	4	5	6	7	8	9	10

4. I am not so burdened by current work load and pressures that it would be hard for me to assimilate the skills needed to make Good Stuff happen.					I am very burdened by other pressures and work load.				
1	2	3	4	5	6	7	8	9	10

5. I have been integrated into the load decision making process about projects.					Most of the decisions are still being made above me without much input from me.				
1	2	3	4	5	6	7	8	9	10

5. The boss asks us to give highest priority to making software that is easily maintainable.					The boss asks us to get software out the door Yesterday.				
1	2	3	4	5	6	7	8	9	10

6. We have a Usability Lab that gathers feedback from inexperienced users. Feedback gets factored rigorously into the product.					We use shorthand codes or message numbers to display errors. If it works, that's good enough.				
1	2	3	4	5	6	7	8	9	10

7. We prevent bugs by spending 10% of the development budget on walkthroughs.					We fix bugs by having a lot of unpaid and undocumented overtime.				
1	2	3	4	5	6	7	8	9	10

8. We have a department made up of journalism majors that write user documentation.					Developers write user documentation if they have time.				
1	2	3	4	5	6	7	8	9	10

9. When bugs are fixed, we rigorously log them and track them back to the development phase in which they occurred.					We just fix 'em.				
1	2	3	4	5	6	7	8	9	10

10. We routinely have a user on the development team.					Users won't talk to us or don't want to be bothered.				
1	2	3	4	5	6	7	8	9	10

11. We know the cost in dollar terms to repair a line of code. We have this data on a product by product basis.					We just fix 'em.				
1	2	3	4	5	6	7	8	9	10

12. Users present us with specifications that have industry standard diagrams or other formal notations which describe their needs.					Users hand us yellow sheets of paper with sketches and comments and hope we can decipher them.				
1	2	3	4	5	6	7	8	9	10

13. The team routinely uses CASE tools and thought before any code is written.					The team takes the yellow sheets and starts making screens and code from them right away. Hey, get real, coding is hard work, we don't have time for that CASE stuff!				
1	2	3	4	5	6	7	8	9	10

14. Old versions of code are routinely managed by a formal version control and library system.					Old versions of code are kept on USB flash drives somewhere, maybe on Charlie's desk.				
1	2	3	4	5	6	7	8	9	10

15. Code modules have part numbers.					Huh? We don't even have any naming conventions for modules or components.				
1	2	3	4	5	6	7	8	9	10

16. Everybody knows what cohesion in code is and frequently agrees on the level of cohesion a module has.					Nobody ever talks about it.				
1	2	3	4	5	6	7	8	9	10

17. We have a formal tuning and optimizing phase built into the development schedule where the system performance is tweaked.					We pride ourselves on being able to write tight code right from the beginning of the project.				
1	2	3	4	5	6	7	8	9	10

18. Upper management believes and respects my time and budget estimates.					Upper management always tries to renegotiate or shorten my estimates under any circumstances (or they have actually cut it recently).				
1	2	3	4	5	6	7	8	9	10

19. We regularly review our software development techniques, bug reports and costs to determine better ways to do things.					We do not even have a formal software development process that all projects follow.				
1	2	3	4	5	6	7	8	9	10

20. I have the authority and upper level support I need to get my work done					Upper management seems distant or uninterested in my problems.				
1	2	3	4	5	6	7	8	9	10

1.