One Flew Over the Cuckoo’s Cubical

What attracted you to software engineering as a profession? Money, fame, toys, independence, intellectual challenges?

What did you find? Unrealistic deadlines, clueless leads, long pointless meetings, carpal-tunnel syndrome, cut-throat colleagues, life as a social outcast?

What profession adopts Dilbert as its poster boy? Why do we passionately defend our profession as an engineering discipline and then genuflect to cowboys, hackers, Bill Gates, and a cartoon character with no life and a bent tie?

I recently participated in a brainstorming session to determine the best software innovation of the 20th century. Three out of four answers were actually hardware innovations. What profession prides itself in its innovation and then articulates that innovation in terms of another profession?

What profession wants its lasting legacy to be Y2K?

Are we crazy? Ken Kesey’s novel One Flew Over the Cuckoos Nest may give us a parallel. Through the Big Chief’s eyes we gain an interesting perspective on conformity and the dangers of being socially pigeonholed. A cast of crazy characters, as sane as you and I, choose to exist within the confines of insanity. For a moment Jack P. M. McMurphy exposes them to freedom, choice, and life outside. In a shocking end McMurphy dies; Taber, Harding, Ellis, and Martini return to the comforts of conformity; and Chief flies the coop.

Software engineers follow an eerily similar path, caught in a struggle between the software we must tame and the monsters we have created to do so. We are no more insane than our neighbors; however, thanks to Dilbert, Y2K, and our inability to meet any schedule or budget, we appear to reside, rather than work, in the asylum. What is most disturbing is that like Taber, Harding, Ellis, and Martini, we seem not to care. In fact, we revel in such images. If you question that, take a stroll around a software engineering conference. It’s not so much about dress as it is attitude. We want to be respected, but only on our terms. We have set a dangerous professional precedence.

In Introduction to Psychology, 8th Edition, Stanford researcher David Rosenhan and 10 colleagues were admitted to a psychiatric hospital by pretending to hear voices. After their admission they acted completely normal. The staff was not suspicious. The pseudo-patients were seen by the staff in the context of a mental ward and labeled schizophrenic. Anything they did was viewed as part of their illness. When they came clean and explained they were faking and not crazy, staff members diagnosed them with paranoid delusions.

Resembling Rosenhan’s group, software engineers have admitted themselves into the cubical asylum. In doing so we have pigeonholed our profession into a category that does not necessarily lead to our professional hopes or dreams. Money, respect, and responsibility seem rather scarce while deadlines, clueless leaders, meetings, and prejudice are in abundance. We want out, but our appeals, like Rosenhan’s, fall on deaf ears. Owners, managers, and customers alike ask how long we have been having these delusions.

What did you find? Unrealistic deadlines, clueless leaders, long pointless meetings, carpal-tunnel syndrome, cut-throat colleagues, life as a social outcast?

What profession adopts Dilbert as its poster boy? Why do we passionately defend our profession as an engineering discipline and then genuflect to cowboys, hackers, Bill Gates, and a cartoon character with no life and a bent tie?

I recently participated in a brainstorming session to determine the best software innovation of the 20th century. Three out of four answers were actually software innovations. What profession prides itself in its innovation and then articulates that innovation in terms of another profession?

What profession wants its lasting legacy to be Y2K?

Are we crazy? Ken Kesey’s novel One Flew Over the Cuckoos Nest may give us a parallel. Through the Big Chief’s eyes we gain an interesting perspective on conformity and the dangers of being socially pigeonholed. A cast of crazy characters, as sane as you and I, choose to exist within the confines of insanity. For a moment Jack P. M. McMurphy exposes them to freedom, choice, and life outside. In a shocking end McMurphy dies; Taber, Harding, Ellis, and Martini return to the comforts of conformity; and Chief flies the coop.

Software engineers follow an eerily similar path, caught in a struggle between the software we must tame and the monsters we have created to do so. We are no more insane than our neighbors; however, thanks to Dilbert, Y2K, and our inability to meet any schedule or budget, we appear to reside, rather than work, in the asylum. What is most disturbing is that like Taber, Harding, Ellis, and Martini, we seem not to care. In fact, we revel in such images. If you question that, take a stroll around a software engineering conference. It’s not so much about dress as it is attitude. We want to be respected, but only on our terms. We have set a dangerous professional precedence.

In Introduction to Psychology, 8th Edition, Stanford researcher David Rosenhan and 10 colleagues were admitted to a psychiatric hospital by pretending to hear voices. After their admission they acted completely normal. The staff was not suspicious. The pseudo-patients were seen by the staff in the context of a mental ward and labeled schizophrenic. Anything they did was viewed as part of their illness. When they came clean and explained they were faking and not crazy, staff members diagnosed them with paranoid delusions.

Resembling Rosenhan’s group, software engineers have admitted themselves into the cubical asylum. In doing so we have pigeonholed our profession into a category that does not necessarily lead to our professional hopes or dreams. Money, respect, and responsibility seem rather scarce while deadlines, clueless leaders, meetings, and prejudice are in abundance. We want out, but our appeals, like Rosenhan’s, fall on deaf ears. Owners, managers, and customers alike ask how long we have been having these delusions.

What attracted you to software engineering as a profession? Money, fame, toys, independence, intellectual challenges?

What did you find? Unrealistic deadlines, clueless leaders, long pointless meetings, carpal-tunnel syndrome, cut-throat colleagues, life as a social outcast?

What profession adopts Dilbert as its poster boy? Why do we passionately defend our profession as an engineering discipline and then genuflect to cowboys, hackers, Bill Gates, and a cartoon character with no life and a bent tie?

I recently participated in a brainstorming session to determine the best software innovation of the 20th century. Three out of four answers were actually hardware innovations. What profession prides itself in its innovation and then articulates that innovation in terms of another profession?

What profession wants its lasting legacy to be Y2K?

Are we crazy? Ken Kesey’s novel One Flew Over the Cuckoos Nest may give us a parallel. Through the Big Chief’s eyes we gain an interesting perspective on conformity and the dangers of being socially pigeonholed. A cast of crazy characters, as sane as you and I, choose to exist within the confines of insanity. For a moment Jack P. M. McMurphy exposes them to freedom, choice, and life outside. In a shocking end McMurphy dies; Taber, Harding, Ellis, and Martini return to the comforts of conformity; and Chief flies the coop.

Software engineers follow an eerily similar path, caught in a struggle between the software we must tame and the monsters we have created to do so. We are no more insane than our neighbors; however, thanks to Dilbert, Y2K, and our inability to meet any schedule or budget, we appear to reside, rather than work, in the asylum. What is most disturbing is that like Taber, Harding, Ellis, and Martini, we seem not to care. In fact, we revel in such images. If you question that, take a stroll around a software engineering conference. It’s not so much about dress as it is attitude. We want to be respected, but only on our terms. We have set a dangerous professional precedence.

In Introduction to Psychology, 8th Edition, Stanford researcher David Rosenhan and 10 colleagues were admitted to a psychiatric hospital by pretending to hear voices. After their admission they acted completely normal. The staff was not suspicious. The pseudo-patients were seen by the staff in the context of a mental ward and labeled schizophrenic. Anything they did was viewed as part of their illness. When they came clean and explained they were faking and not crazy, staff members diagnosed them with paranoid delusions.

Resembling Rosenhan’s group, software engineers have admitted themselves into the cubical asylum. In doing so we have pigeonholed our profession into a category that does not necessarily lead to our professional hopes or dreams. Money, respect, and responsibility seem rather scarce while deadlines, clueless leaders, meetings, and prejudice are in abundance. We want out, but our appeals, like Rosenhan’s, fall on deaf ears. Owners, managers, and customers alike ask how long we have been having these delusions.

What attracted you to software engineering as a profession? Money, fame, toys, independence, intellectual challenges?

What did you find? Unrealistic deadlines, clueless leaders, long pointless meetings, carpal-tunnel syndrome, cut-throat colleagues, life as a social outcast?

What profession adopts Dilbert as its poster boy? Why do we passionately defend our profession as an engineering discipline and then genuflect to cowboys, hackers, Bill Gates, and a cartoon character with no life and a bent tie?

I recently participated in a brainstorming session to determine the best software innovation of the 20th century. Three out of four answers were actually hardware innovations. What profession prides itself in its innovation and then articulates that innovation in terms of another profession?

What profession wants its lasting legacy to be Y2K?

Are we crazy? Ken Kesey’s novel One Flew Over the Cuckoos Nest may give us a parallel. Through the Big Chief’s eyes we gain an interesting perspective on conformity and the dangers of being socially pigeonholed. A cast of crazy characters, as sane as you and I, choose to exist within the confines of insanity. For a moment Jack P. M. McMurphy exposes them to freedom, choice, and life outside. In a shocking end McMurphy dies; Taber, Harding, Ellis, and Martini return to the comforts of conformity; and Chief flies the coop.

Software engineers follow an eerily similar path, caught in a struggle between the software we must tame and the monsters we have created to do so. We are no more insane than our neighbors; however, thanks to Dilbert, Y2K, and our inability to meet any schedule or budget, we appear to reside, rather than work, in the asylum. What is most disturbing is that like Taber, Harding, Ellis, and Martini, we seem not to care. In fact, we revel in such images. If you question that, take a stroll around a software engineering conference. It’s not so much about dress as it is attitude. We want to be respected, but only on our terms. We have set a dangerous professional precedence.

In Introduction to Psychology, 8th Edition, Stanford researcher David Rosenhan and 10 colleagues were admitted to a psychiatric hospital by pretending to hear voices. After their admission they acted completely normal. The staff was not suspicious. The pseudo-patients were seen by the staff in the context of a mental ward and labeled schizophrenic. Anything they did was viewed as part of their illness. When they came clean and explained they were faking and not crazy, staff members diagnosed them with paranoid delusions.

Resembling Rosenhan’s group, software engineers have admitted themselves into the cubical asylum. In doing so we have pigeonholed our profession into a category that does not necessarily lead to our professional hopes or dreams. Money, respect, and responsibility seem rather scarce while deadlines, clueless leaders, meetings, and prejudice are in abundance. We want out, but our appeals, like Rosenhan’s, fall on deaf ears. Owners, managers, and customers alike ask how long we have been having these delusions.

What profession adopts Dilbert as its poster boy? Why do we passionately defend our profession as an engineering discipline and then genuflect to cowboys, hackers, Bill Gates, and a cartoon character with no life and a bent tie?

I recently participated in a brainstorming session to determine the best software innovation of the 20th century. Three out of four answers were actually hardware innovations. What profession prides itself in its innovation and then articulates that innovation in terms of another profession?

What profession wants its lasting legacy to be Y2K?