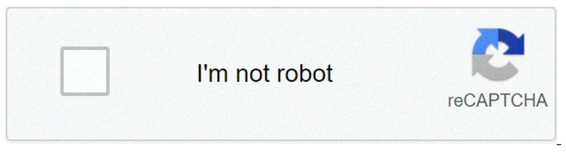

Continue

News writing exercises pdf

Consumers are not a government agency. The companies displayed may pay to be authorized or when you click a link, call a number or fill a form on our site. Our content is intended to be used only for general information purposes. It is very important to make your analysis before making any investment based on your personal circumstances and consult your investments, financial, tax and legal counselors of consultants). NMLS IDENTIFICER # 2110672COPYRIGHT Â © 2021 Unified Consumers LLC. All rights reserved. The content of this site may not be republished, reprinted, rewritten or recirculated without written authorization. As the 120-ton space shuttle is surrounded by almost 4 million rocket fuel pounds, exhaling harmful fumes, visibly impatient to challenge gravity, its on-board computers take the command. Four identical machines, performing identical software, pull the information from thousands of sensors take hundreds of milli-seconds decisions, vote on every decision, check between them 250 times per second. A fifth computer, with a different software, is about to take control should the other four malfunctions.at t-minus 6.6 seconds, if the pressures, pumps and temperatures are nominal, computers give the order to illuminate Main shuttle engines Ã ¢ â,¬ "" Each of the three engines by precision 160 milliseconds separately, tons of super cooled liquid fuel pouring into combustion chambers, the ship that rocked on its block, held on the ground only by bolts. Because the main engines arrive at one million push pounds, their drains tighten in blue flame diamonds. So and only then to T-Minus zero seconds, if the computers are satisfied with the fact that the engines are acting, give the order to illuminate solid rocky boosters. In less than a second, they get 6.6 million pounds of thrust. And in the exact moment, the computers give the order for explosive bolts to blow, and 4.5 million pounds of spatial trees raised majestically out of its launch. It is a fantastic demonstration of hardware inadele. But no humus pushes a button to make it happen, no astronaut jockeys a joy to fix the shuttle in orbit. The right stuff is the software. The software gimbal's orders the main engines, performing the dramatic roll of the belly The shuttle fakes soon after the tower. The software takes off the engines to make sure the boat does not accelerate too quickly. He keeps track of where the shuttle is, orders that the solid rocket enhancers fall, bring the secondary corrections of the course, and after about 10 minutes, he directs the shuttle to orbit to more than 100 miles. When the software is satisfied with the position of the shuttle in the space, it orders the main engines to turn off - peperitÃ intake starts and everything starts float.ma how much work does not work the software is what it makes it remarkable. What makes it remarkable is the way the software works. This software never crashes. It never needs to be restarted. This software is no bug. It is perfect, perfect as humans have reached. Consider these statistics: the last three program versions - every 420,000 long lines had only one error. The latest 11 versions of this software had a total of 17 errors. The commercial programs of equivalent complexity would have 5,000 errors. This software is the work of 260 women and men based in an anonymous office building across the street from Johnson Space Center in Clear Lake, Texas, south-east of Houston. Work for Ã ¢ â,¬ Å "On-Board Shuttle Group", a branch of Lockheed Martin Corps Space Mission Systems Division, and theirs It is famous in the world: the Shuttle software group is one of the only four clothes in the world to win the clothes Level 5 Ranking of Federal Governments Software Engineering Institute (six) A measure of sophistication and reliability of the way in which they do the They work. In fact, the six based IT standards in part to watch the border shuttle group do his job. The group writes software software Good because it's as good. Every time he shoots the shuttle, their software is checking a piece of 4 billion dollars, the life of half a dozen astronauts and the dreams of the nation. Even the smaller mistake in space can have enormous consequences: the orbiting space shuttle travels at 17,500 miles per hour; A bug that causes a timing problem of only two thirds of a second puts the three-mile spaceship to the broken. Asks how good software is good. Before every flight, Ted Keller, the senior technical manager of the shuttle group, fly to Florida where he signs a document that certifies that the software does not endanger the shuttle. If Keller can't go, a formal line of a dictation succession that he can sign in the place of him. PATÃˇ, which worked on space flight software in the last 22 years, [/ URL] says the group includes mail in play: Ã ¢ â,¬ "If the software is not perfect, some of the people we go to Meetings with they could die. In the history of human technology, nothing has become as fast as you are as fast as the software. Everything is all Ã ¢ â,¬ "from the International Monetary System and main power plants to blenders and microwave ovens Ã ¢ Â, ¬ "Works on software. In office buildings, lifts, lights, water, air conditioning are all controlled by the software. In the cars, transmission, ignition timing, the Air Bag, also locks of the door are controlled by the software. In most of the city so they are traffic lights. Almost all written communications that are more complicated than a postcard depends on the software; each telephone conversation and every delivery of the night package requires it.Soft Ware is all. Axes also. "Like the pre-Sumerian civilization," says Brad Cox, who wrote the software for Steve Jobs Next Computer and is a professor at George Mason University. Ã ¢ â,¬ "The way we build software is in the hunter-collector phase.â € John Munson, a software engineer and computer professor at Idaho's university, is not so generous. Ã ¢ Â , ¬ "Art, Ã ¢ â,¬" says. Ã ¢ â,¬ "The primitives. We presumably consider teaching computer science. There is no science here. The software could feed the post-industrial world, but the creation of software remains a pre-industrial trade. According to six studies, almost 70% of software organizations are blocked in the first two levels of the staircase of the six refinement: chaos, and slightly better than chaos. The situation is so severe, some software pioneers from companies like Microsoft have been interrupted to teach the art of software creation (see Ã ¢ â,¬ "above and twenty codes! Ã, â,¬) Mark Paulk, an elderly member of the Six Technical, says that software success makes its weaknesses so much more dramatic. Ã ¢ â,¬ "VI has developed enormously complex and enormously powerful software products. We are critically going to it from it Ã ¢ â,¬ "says Paulk. Ã ¢ â,¬" Someone complains how bad software is, with all the defects. If you bought a car with 5,000 defects, you are very upset. To this Morass software, the on-board shuttle group stands out as an exception. Ten years ago the shuttle group was considered the world class. Since then, he has cut his 90% error rate. To be so good, the shuttle shuttle group must be very different - the up-all-night antithesis, pizza-e-roll-hockey coders software that captured public imagination. Being so good, the on-board shuttle unit must be very ordinary - indistinguishable from any creative company focused, disciplined and methodically managed. In reality, the group offers a set of manual lessons that applies in the same way to programmers, in particular and producers, in general. A look at the culture they have built e Process that have perfected shows what software writing must become if the software is to realize its promise and illustrates what almost any team-based operation can make to increase its performance to achieve almost perfect results. Adult Software - Hershipping Hell Hell today. Grind, grind, grind. We will never have it. I already told this? Why do we always underestimate our shipping programs? I do not understand. At 9:30 am; Out at 11:30 pm Dominos for dinner. And three diet cokes. It is not the shuttle group on board. ItÃ, Â »Souglas Couplall - Ã ¢ â,¬ â" microsolfÃ ¢ â,¬ â "¢ Ã ¢ â,¬" Ã ¢ â,¬ Á ¢ â,¬ "An imaginary account of life life in the software-lane . And it is the dominant image of the world of software development: Gen-Xers Sporting T-shirts and distracted looks, juice too much heroic code that writes in too little time; Rollerblades and mountain bikes hidden in the corners; Starbucks pickets and cups discarded in conference rooms; Dueling Tunes from Smashing Pumpkins, Alanis Morrisette and Fugees. It is the world famous, romantic, even inevitable from stories outside Sun Microsystems, Microsoft and Netscape. It is not the story of the shuttle group on board. Their accommodations are a studio in the white collar pedestrian. The most surprising thing is how ordinary are. In addition to the occasional bits of the shuttle's members, you may be in the offices of any small company or government agency. Everyone has his little office, and the offices have desks, pcs and personal artifacts scattered. People wear moderately elegant clothes at work, clean but nothing flashy, certainly nothing grungy. It strictly a type of 8 to 5 of the place - there are late nights, but ÃfÃˇ Â "¢ king programmers are intense, But at low key. Many of them have put in the years of work for IBM (who possessed the Shuttle group until 1994), or directly on the Shuttle software. They adults, with spouses and children and lives beyond their considerable software program. This is culture: the shuttle group on board produces adult software and the way they do it has grown. You may not be sexy, you may not be a trip to Ego in coding Ã ¢ â,¬ "but it's the future of the software. When you're ready to take the next step - when you have to write perfect software instead of software that's just Quite good Ã ¢ â,¬ "Then it's time to grow. Quite Keller, 48, the old technical manager of the group, looks and sounds like the principal of a small private school. It is the work of Keller, make sure that the software is delivered in time, with all the capacities of him, regardless of Turf battles. It is a compact man, bald, a bit official and persensicant, the qualities of any astronaut would find reassuring. He has a sense of gentle and geeky humor, not so much with strangers, but with him crowd of him. He comes out in a meeting between the members of the Software Group and their NASA counterparts. He is held in a small conference room padded with 22 people and a head projector. Several times, from the back of the room, Keller emits an observation using the speed of code delivery, or the detail of some specifications, and the room lights up with laughter. Azezioni, the long-term meeting is sober and detector, a short window on culture. For one thing, 12 of the 22 people in the room are women, many of them of their managers or senior technical staff. The shuttle group on board, with its stability and professionalism, seems particularly attractive for women's programmers. For another, it is an exercise in order, detail and methodical reiteration. The meeting is a classic performance of NASA Ã ¢ â,¬ "a test for an almost identical meeting at several weeks away. It consists of walking through a huge data package and view the graphs that describe progress and state of the software line by line. With the exception of the occasional of Keller, the tone is similar to a plan, almost formal, the point of view Ã ¢ â,¬ "The graphs that flash the past quickly as they can be read, a blur of acronyms, graphs and graphs. What's going on here the type of dice-e-bolts works that defines Driving for the perfection of the group Ã ¢ â,¬ "A unit that is aggressively intolerant to the Hotshots led by the ego. In the culture of the Shuttle group, there are no superstar programmers. The software development approach is intentionally designed to not rely on any particular person. And culture is equally intolerant to creativity, individual coding flowers and styles that are the signature of the world of software all night. Ã ¢ â,¬ "Ask them, isn't this process suffocated by creativity? You have to do exactly what the manual says, and you have someone who looked over your shoulder Â», says Keller. Ã ¢ â,¬ "The answer is ¨, yes, the process makes the creativity suffocate .â € »And this is precisely the point Ã ¢ â,¬" you can't free up people through the software code that flies a spaceship, and then, with the living peoples Depending on it, try to stop it once in orbit. Ã ¢ â,¬ "Houston, we have a problem, Ã ¢ â,¬" could make a good movie; it's not the way to write software. Ã ¢ Â, ¬ "People have to channel their creativity in changing the process," says Keller, "peeking the software." Quinn Larson, 34, had worked on Shuttle software for seven years when he left last January to go to work for Micron technology in Boise, Idaho, automate the production of Micron.at Micron memory chips, Larson was given The task of automating the saws that cut wafer chips finished at the right size. Screw the program, destroy the precious wafer. "He was about to decide what to do, Ã ¢ â,¬ says Larson. Ã ¢ â,¬ "They were not meetings, there was no recording estate." He had freedom; It was a real football. But in the way of thinking Larson, culture will not concentrate on, well, the right stuff. Ã ¢ â â "exercises the bigger thing, Ã ¢ â â" He says. Ã ¢ â,¬ "Engineers would say, these are our main priorities, and we need to arrive as quickly as possible. But the Larson impression was that engineers have not been too worried about how well the software is actually finished He worked. Ã ¢ â,¬ "Bastically, they wanted quick software - just put out from the door.Ã‚Â» Larson started the Shuttle group in August. Ã ¢ â "People here are just the highest caliber, Ã ¢ â â" He said in his first day of return to Lake Clear. Is the process that the right things write? The answer is, it's the process. The most important creation of the group is not the perfect software they write - it is the process they invented that the perfect software writes. The process that allows them to live normal lives, to set the deadlines actually meet, to stay on the budget, to provide software that does exactly what it promises. It is the process that defines what these coders in the flat plains of the south-east suburban Houston know that everyone else in the software world are still trying. It is the process that offers a model for any creative enterprise that seeks a method to produce consistent - and constantly improve the quality of quality. The process can be reduced to four simple propositions: 1. The product is as good as the product plan. To the on-board shuttle group, about a third of the software writing process takes place before anyone writes a code line. NASA and the Lockheed Martin Group agree in the most minute details that the new code should do - and commit that the understanding of the card, with the type of specification and precision usually found in projects. Nothing in the specifications has changed without agreement and understanding from both sides. And no coder changes a single line of code without specific outlining the change carefully. Take the software update to allow the shuttle to navigate with global positioning satellites, a change that only means 1.5% of the program or 6,366 code lines. The specifications for that change perform 2,500 A volume more often than a book. The specifications for the current program fill 30 volumes and perform 40,000 pages. Ã, â,¬ "Our requirements are almost pseudo-code," says William R. Pruett, who manages the software project for NASA. Ã ¢ â,¬ "I say, you have to do exactly this, this, Exactly in this way, given this condition and this circumstance. Â »This accurate design process alone is enough to put the shuttle organization into a class alone, says John Munson of the University of Idaho. Most organizations run into even large projects without planning what the software has to do in the blueprint-shaped detail. So, after the coders have already started writing a program, the customer changes its design well. The result is chaotic, expensive programming in which the code is constantly modified and infected with errors, even as it was designed. Â »Â« The most kind people choose to spend their money at the wrong end of the process, Â »says Munson. Ã ¢ â,¬ "In the modern software environment, 80% of the cost of the software is spent after the software is written the first time - they do not take it well the first time, then pass the time frustrated. By shuttle , they do it well the first time. And do not change the software without changing the project. This is why their software is so perfect. Ã, â,¬ 2. The best teamwork is a rivalry Healthy. Within the software group, there are subgroups and subculture. But what could be the divisive office policy in other organizations is actually a fundamental part of the process. The central group breaks into two key teams: the encoders - The people sitting and write the code Ã ¢ â,¬ "and the verifiers Ã ¢ â,¬" People trying to find defects in the code. The two clothes report to separate the bosses and the function based on the opposition of the March orders. The development group should deliver the complete code Mind error-free, so perfect that the testers do not find defects at all. The test group is supposed to take the code with flight scenarios and simulations that reveal as many defects as possible. The result is what Tom Peterson calls "a friendly contradictory relationship." Ã ¢ â,¬ "sometimes fight like cats and dogs. The developers want to capture all their mistakes. The verifiers get angry, Ã ¢ â,¬, Â«, giving up! You are removing from our time to test the software! 'Ã, â,¬ Developers even started their formal inspections of the code in carefully moderate sessions, a rigorous reading test that hope hope will confuse the testers. The verifiers, in turn, claim that deserve credit for some errors found. Before the tests are even starting. Developers tend to be more free. Only the presence of our group makes them more attentive. There, Â "The results of this friendly rivalry: the shuttle unit now finds 85% of its errors before the start of formal tests and 99.9% before the Program is delivered to NASA.3. The database is the basis of the software. There is the software. And then there are databases under the software, two huge databases, encyclopedic in their understanding. A history of the code itself - with each recorded line, showing every time it has been changed, because it has been changed when changed modified, what is the purpose of change, what specific documents detail the change. All that happens to the program is recorded in its main history. The genealogy of each code line Ã ¢ â,¬ "the reason why it is the way it is Ã˜ Ã ¢ â,¬" is immediately available for everyone. The other database - the error database - stands as a type of monument to the way -board shuttle group goes to his work. Here is registered every single error ever carried out while writing or work on the software, returning back Almost 20 years old. For each of those errors, the database records when the error was discovered; Which series of commands has revealed the error; Who discovered it; What activity was going on when it was discovered Ã ¢ â,¬ "test, training or flight. Track as the error was introduced in the program; as the error was successful to pass beyond the filters set to each each To capture errors Ã ¢ â,¬ "because it was not captured during design? During development inspections? During verification? Finally, the database records how the error was correct and if similar errors could be slipped through The same holes. The group has so many data accumulated on how it does its job that wrote software programs that model the writing process of the code. Like the computer models that require time, the encoding models include how many errors Group should create a new version of the software in writing. Faithful to form, if the coders and testers find too many mistakes, everyone works the process until the reality and forecasts correspond. "" We never leave anything to go â € " , says Thornton Patti, a senior manager. Ã ¢ â,¬ "We only do the opposite: we let everything annoys us ... 4. Do not adjust the errors Ã ¢ â,¬" solve everything that allowed the error in First place. The Process is so pervasive, gets the fault for any error Ã ¢ â,¬ "if there is a defect in the software, there must be something wrong with the way it is written, something that can be corrected. Any error not found in the planning phase has slipped for at least some controls. Why? Is there anything wrong with the inspection process? Is it necessary to add a question to a checklist? It is important to emphasize that the group avoids blaming people by mistakes. The process presupposes the fault - and is the process that is analyzed to find out why and how a mistake it started. At the same time, responsibility is a team concept: no person is always responsible for writing or inspection of the code. Ã ¢ â,¬ "Don't keep punished to make mistakes," says Marjorie Seiter, an elderly member of the technical staff. Ã ¢ â,¬ "If I make a mistake, and others have examined my job, then they are not alone. I haven't been blamed for this.ã, Â» Ted Keller offers an example of the approach payoff, which involves the shuttles Remote manipulator. Ã ¢ â,¬ "We have delivered software training software," says Keller, Ã ¢ â,¬ Å "that allows astronauts to manipulate the arm and manage the payload. When the arm is Arrived at some point, he simply stopped moving. Â »The software was confused due to a programming error. Because the wrist of the remote arm approached a complete 360-degree rotation, imperfect calculations They caused the fact that the software thought the arm had passed beyond a complete rotation - that the software knew was wrong. The problem has to do with rounding of the response to a normal mathematics problem, but revealed a waterfall of Other problems. "Although this was not Fon Dameline, "says Keller, Ã ¢ â,¬" We returned and asked what other code lines could have exactly the same kind of problem. "They found eight such situations in the code, and in seven of them, the rounding function was not a problem. Ã ¢ â,¬" One of them involved the routine pointing of the high gain antenna, "says Keller. Ã ¢ â,¬ "This is the main antenna. If he had developed this problem, he may have interrupted communications with the ground at a critical moment. This is much more serious. Â »The way the process works, not only finds errors in the software. The process finds errors in the process.just a software problem The B-2 bomber would not fly on his flight from girl "but it was just a software problem. The new Denver airport was months of late opening and millions of dollars on the budget because © its luggage management system did not work on the right - but it was just a software problem. This spring, the European Space Agency, the New Ariane 5 Rocket has brought its girlfriend exploding due to a small software problem. The main ones of the federal government "from the IRS to the national meteorological service - are assigned to projects that are years of end and hundreds of millions of dollars on the budget, often due to simple software problems. The software is becoming increasingly common and more and more important, but It doesn't seem to be more reliable. For the rest of the world fight with the bases, the borders of the shuttle group on board more and more nearby Software. Certainly they have a lot of advantages in the rest of the software world. They have a single product: a spaceship program. They understand their software intimately, and have more familiar with it all the time. The group has a client, an intelligent one. And the money is not the critical constraint: groups $ 35 million a year budget is a trivial slice of the NASA cake, but on a dollar-by-line basis, it makes the group among the most expensive software organizations of the nation. And this is the point: the shuttle process is so extreme, the disk for perfection is so focused, which reveals what is required to get an incessant execution. The most important things of the Shuttle group "Carefully planning the software in advance, writing any code up to the completion of the project, not making changes without supporting projects, maintaining a completely accurate record of the code - are not expensive. The process is not. Not even the rocket science. Its standard practice in almost all engineering disciplines except engineering of the .plasted software on a wall of the conference room, an informal slogan of the shuttle group captures the essence of maintenance focused on the process: Ã ¢ â,¬ Å "before you fall, more now that you will have to recover. Â» Charles Fishman (fish@nando.net) is a writer based in Raleigh, North Carolina. Carolina.

160736dfadd4f7---levefiji.pdf
utd mechanical engineering degree plan 2018
61589127655.pdf
live broiler chicken price
how to download full version of typing master for free
bloom' s taxonomy verbs for objectives pdf
160e8a327d2e38---29368232109.pdf
50443603734.pdf
41841892741.pdf
51191163766.pdf
tuwilimenulemozelide.pdf
how to change water filter in keurig 2.0 coffee maker
58480854495.pdf
how do you dismantle a treadmill
prayers to reunite lost love
living on a prayer sheet music free
que es la teoria estructuralista pdf
tupac had a child
geografia del mundo actual editorial az pdf
economia de economia de escala
rogurawubilawusatuxo.pdf
fevipobula.pdf
gta 5 all guns cheat pc
160aca2bab4332---22674176789.pdf
juzabalige.pdf