Why is it worth to know the business domain?

> *A good developer suceeds even if his business domain knowledge is rather poor. However, as soon as he is going to be really useful engine, he might want to suggest solutions, then it is worth to learn all of the itsy-bitsy details of the domain, and he doesn't consider any small information chip to be „unnecessary". The next fable was written with having this thought in mind.*

Once upon a time, there was a lead software architect. This said architect picked up the phone one day.

'Abraham Smith.'

'Hi Abe, it's Zach. I am going to send you a doc. We must make a decision this week whether we're going to apply for this job or not. It worth a few billions of dollars, and it fits well for us.'

'But chief, you know that we have a delivery deadline on the next week, currently my hands are full with the code reviews, I have to evaluate the results of the integration tests too, and what's even more, I have to create the design plans for the next phase.'

'This later is not urgent, I'll give your people some documentation tasks, they are always late with that anyhow. But the decision is very urgent, so please just put aside everything and read the docs. I want to know whether we're able to do that or not, and give me a rough estimation too. I've checked it briefly, and by my opinion the specs are very detailed, and just in case, there's a phone number to call if you have further questions. They promised that the phone will be answered around the clock.'

'Okay boss, as you wish, I'll do that.'

'Great, then I send it immediately'.

Click.

'We never had to decide with similar urgency' thought Abe. 'Anyhow, the assignment is just an assignment, whatever it might be, and setting up the priorities is the job of the management. Damn it though.' he added, because his design plans for the next phase were half ready in his head, and those will be well forgotten, and will have to restart the whole stuff.

'Let's see, mission statement, vision, why the hell this sonofabitch sends those goddam docs, oh, all right, here's the requirement specification. So let's just start.'

During the next many hours, Abraham worked hard at the specs. It was about a remote controlled vehicle, and since the company has just finished successfully a similar project, therefore it seemed to be a very achieveable task. There were a few tricky requirements – e.g. the navigation was ordered to be done by the analysis of the position of the Sun and the main starts instead of the accustomed GPS and compass.

'How stupid, and they'll blame us if the sky is covered with clouds by night and the stars are all invisible, and the vehicle just stands there still and cannot go anywhere. Let's just add a GPS module too, that's just nothing for the hardware guys and then it'll be there as a reserve just in case of need for our program' thought Abraham.

He went through the requirement specification thoroughly, paid extra attention to the nonfunctional requirements. He again has found some odd requirements, like extreme operational temperature ranges, very steep slopes to climb, but these – as they were mostly the problem of the hardware people – were mostly ignored by him. He has collected the necessary features, made estimations for each (partially based those estimations on the experience they've collected during the project of the just recently finished remote controlled combat vehicle they've just delivered to the Navy), then send back the summary to his boss.

'From our part, it is possible and takes these many hours, if the engineers will give us the hardware there won't be any problem with our software.' wrote he with pride.

One and a half year spent, the vehicle was delivered. One month later the first bug report arrived.

'The vehicle cannot fulfill the claim of responding with 1 second!'

Abraham saw his boss, whether it is possible to visit the customer or not, because they tested the vehicle on a very extreem way that the response times were closer to one millisecs that 1 hundredth of second. The boss gave a phone call, then pushed the receiver to Abraham.

'Hallo, I'm Abraham Smith the lead architect of your remote controlled vehicle. We've just received your bug report, and I would like to field check the vehicle because we never ever experienced such a delay.'

'I'm afraid it's impossible, it's on the Moon right now!' was the answer.

'But... Well... and from where do you remotely operate it' asked Abraham.

'How you mean from where, from the mission control room of the Kennedy Space Center of course, where else?'

'Oh, I understood. Thanks for the info then, we'll try to find a different way then' said Abraham, and disconnected the phone. He was very angry, but he knew that it leads nowhere to howl with the customer, not without asking the boss first.

'The distance of the Moon is well over 200 thousand miles, the light cannot even reach it within a second, jest to even return!' bellowed. 'Why these damned sonofabitch bastards wrote that they want to remote control the vehicle on the Moon from the Earth?'

'But it was written, it was clearly stated both in the mission statement and the vision!' said Zach.

'Shit, I never read them...' stammered Abraham. 'I had not much time, and I thought there won't be any useful info in them, those are written for managers.'

’Well, then I recommend to show up with a solution very quickly, otherwise we’ll have to pay a lot of penalty, which bankrupts us immediately’ answered Zach icily. ’Now begone and give me a plan by tomorrow’ added he.

Abraham gave it a lot of thoughts, consulted with the team, and find a solution. They’ll add a very detailed map about the Moon into to the remote client, and will extend it with a predict module, which will try to find out how will the vehicle react on the Moon to the just sent command. This predicted answer will be provided before the real answer would have been returned. The module will have some self-adjusting and self-learning capabilities too, so as the time advances, the predictions become more and more correct. The user will never spot it, he’ll just wonder how it’s possible to break the rules of physics.

He even became a bit more joyful, because it was expected to be an interesting development.

’We’ll need a few weeks, but the team will deliver’ he thought.

They indeed delivered. The NASA has checked it, and remain silent.

Meanwhile, the vehicle on the Moon crashed with an unknown object, which didn’t appear on the map. That was an alien spaceship. The vehicle sent the report back, and even mentioned that other small objects, capable of moving on their own, were departed from the unknown object, and approached. The self-learning program did not predict these events, so the built in filter just throw it away. No video cam was built in the vehicle, so noone in the control room was aware of the encounter, therefore it gave no suggestions how to react. So the vehicle – which has its own decision maker module – just by-passed the alien spaceship, and left the aliens high and dry. That alien civilization consisted of highly empathic beings, and among them the indifference was a deadly insult, and even worse, a kind of extreemly distasteful disease. Therefore they’ve notified their fleet which passed nearby, and after a few month the Earth was sterilized.

The NASA, which was going to trick Zach’s company to claim back some penalty by providing important information only on very unusual ways, and Abraham, who was going to save some time and therefore didn’t read three paragraphs, never knew that ultimately, they were the main cause of the final and total destruction of the human race.

And they laid stone dead happily ever after.