No more green fields

I carry this model of the perfect job in my mind: at the interview I'm told: "we have a few people here, we have a problem and we have no idea of how we are going to solve it". So, maybe I never expect to hear this but I settle for: "our current team has decided to develop a completely new product without using any existing code".

What I want is: green field development, a brand new development, with no old code to maintain, no creaking database scheme, no legacy source code control system.....

In my old age, in my scepticism I don't believe there are any green field projects left. So much has been done that every area has been touched by legacy systems. Sometimes this is obvious: we must keep the current system working; sometimes it is contradictory: "we are writing a new product but we plan to salvage as much as we can from the old one". Even if they were to throw away all the code they would want to keep the user interface, or the database format, or the file format.

Yesterday's models form part of today's reality: people's expectations have changed – 10 years ago a text based calendar system was magic, today people want an easy to use GUI, voice control, and, and....

Your perspective is always shaped by what you know and what has come before; today's problem are shaped by what exists, so, even if a company doesn't have an application installed they will have users accustomed to some systems.

Some of these advances are good: defining an XML schema is better than defining a byte-by-byte file format, but such developments can be limiting: you may need to use XML because it is a buzz word, forget the fact that you are writing Space Invaders, you must use XML somewhere! Models can constrain us too.

It is not only technology which limits us. If you think of your problem domain as a blank canvas, or better, a Christaller like uniform plain it is still bounded:

- Our Easterly edge is compatibility: we must take data from this system, or produce output for that system, we are asked to reuse as much of the existing system as we can – even though we are supposed to replace it!
- A project deadline and costing forms our Westerly edge: maybe we have a drop-dead project deadline, or maybe it simply effects our bottom line! Maybe if we don't deliver in time the company will cease to be. Even worse can be no deadline, an abyss on the Western edge into which our team disappears on a blue-sky mission to seek out and explore strange new technologies.
- To the North there is the advance of technology: our project will not be the first, or the last, system already out of date when delivered; or maybe, you must target 386 Windows 3.1 machines to save on costly upgrades.
- And to the South our co-workers: we usually don't get to choose the people we work with, some are there when we arrive, some are hired without our involvement; some may be fresh from college and lacking in real world experience, others may be jaded by too many failed projects or over exposure to COBAL and Lisp.

And then when the terrain is not flat and fertile:

- Barren patches mean that crops don’t take: we may advance Extreme Programming, templated designs or code reviews but if people are unenthusiastic, and management won’t back you then nothing will grow.
- Quagmires: you advance a generic design, but before you can cut a line of code you are taken at your word and now your design must work for your department and four others. You are bogged down in endless meetings, advocacy, design reviews. Sir Humphrey would be proud!
- The seasons change: in the autumn the company decides to standardise on Oracle for all database work, suddenly you must abandon SQL Server; winter brings a cash flow crisis and nothing will get signed off; spring brings a thaw but suddenly Java is in flavour, and then summer when everyone is on holiday, nothing is decided and even less done!
- And through the centre of the plain runs the rift valley of office politics: will your proposals prove too radical for the management? Would co-operation with the New York office further someone’s career? Does the office secretary hate contractors and make their life difficult? Could you use COM in your design? Is the senior architect convinced he knows all the answers? Is the director’s ear bent by a long standing employee who grew up on Pascal and frankly, doesn’t know the first thing about objects? Sometimes you have to choose your battles.