

# Chess Moves: Avoid Getting Checkmated in Your Project

by Philip Chu

## Table of contents

1 Publication Information.....	2
2 Look Ahead.....	3
3 Keep Cool.....	3
4 Manage Your Time.....	4
5 Plan for the Worst.....	4
6 Do Your Research.....	4
7 Keep Your Ego in Check.....	5
8 Know Your Pieces.....	5
9 The Sacrifice.....	6

## 1. Publication Information

Copyright ©2005-2008 by Philip Chu All rights reserved.



"[Chess is] as elaborate a waste of human intelligence as you can find outside of an advertising agency." - Raymond Chandler, in *The Long Goodbye*

*I often hear and sometimes use sports analogies used for software development and project management (scrum, anyone?). But I think perhaps chess is a better match - I've often gaped at a project decision and thought to myself, anyone who played chess wouldn't have done that!*

## **2. Look Ahead**

One of my peeves - in television and film, the winner of a chess game always announces "Checkmate!" to the other player's surprise. In reality, that only happens when at least one player is so completely inexperienced or incompetent that they can't see the blindingly obvious. A chess game doesn't consist of two players bungling around at random until someone stumbles into a checkmate. Any strategy involves looking ahead at least a few moves, and any minimally decent player is capable of seeing immediate threats. Strong players can plan ahead several moves, and expert players can tell if a subtle weakness in position guarantees their eventual loss. In any case, chess players resign once they recognize the game is lost.

In fact, any decent chess program is based on planning - the further the program can look ahead, the stronger its performance.

It's amazing how often a project is started without anyone figuring out the steps to get there - the team stumbles ahead one move at a time until it's inescapably obvious that the project is late or infeasible.

## **3. Keep Cool**

Chess isn't poker, but it pays to keep your composure. I played one opening so badly I lost my queen almost immediately, but my nonchalance seemed to convince my opponent that it was an intentional sacrifice, and eventually he crumbled. That was on the Iowa high school circuit, but I still as a high-schooler I entered a professional tournament where my adult opponent was so angrily surprised to find himself in a losing position that he just started moving my pieces to demonstrate he knew how I had the game won, then toppled his king. Saved me a lot of work.

In my experience, project managers often behave the same way - lots of flailing and blame.

- Midway through my first government defense contract, my project manager belatedly realized we were in trouble (we started late, received our equipment late, and had no idea what we were doing), blew up at me, offered me the moon if the project succeeded, then immediately retracted it when he realized he couldn't deliver, then just avoided me for the rest of the project because I was steamed. Another boss of mine called me away from final work on a game release to whine for half an hour how hard everything was for her and that I wasn't taking this seriously, then yelled at me to hurry up and stop wasting time. In both cases, the product somehow was ready to go on time, and so was I.

If the outlook is grim or uncertain, you need to emanate confidence and competence. If you've recognized the game is lost, acknowledge it gracefully and move on to the next game.

## **4. Manage Your Time**

So you need to plan ahead, but you also need to balance that against managing your time. In tournament chess play, you have a fixed amount of time to play a certain number of moves. Thus your schedule is flexible, sort of. If you dawdle through the beginning of the game, you'll be pressed to move your pieces quickly later without much time to think about them. Experienced players are familiar with a standard set of opening moves and typically zip through those quickly and can allocate more minutes to the more complex positions later in the game.

I remember one chess game in which I squandered so much time, pondering one move that I had to move immediately, speed-chess style, for all the remaining moves. I managed to eke out a draw, but only because I knew exactly what moves to make without allowing any time for thought and any room for error. I've had quite a few software releases like that, too, where we dawdled, indecisively went down different paths, and in the end pulled mandatory all-nighters to hit a (in some cases immovable) release date. In some cases, you could say the result was a draw. In other cases, our chess clock ran out. I remember working on one investor party demo even after the party had started - by the time we were ready to show it, everyone had finished their drinks and left.

## **5. Plan for the Worst**

Another TV cliché - a person walks over to someone's in-progress chess game, takes a look and moves a piece, announcing "Checkmate!" Again, this never happens. Unlikely as it is for a player not to see himself losing in the next move, it is even more unlikely that the other player can't see the mate in one, either. It's like a prize fighter not seeing his opponent drop his guard. Chess players plan their strategies assuming their opponents will make the best possible moves they can see. In fact, that's how chess programs are implemented.

In a project, you don't have to assume that the worst will happen, but you should plan for it. It's called risk management. You can't open yourself to the worst case and then forget about it. This is why you have insurance, this is why you don't blow your life savings in Vegas.

## **6. Do Your Research**

A player at a tournament in Des Moines offered to show me his can't-lose opening. After I decimated him in a few moves, he said, wait, I've got another one. Again, a losing proposition. If there was such a thing as a guaranteed winning opening, the game would be no more interesting than tic-tac-toe. There's a reason for all those encyclopedic compendiums of chess openings and endings - a lot of chess experts over the years have tried them out and done the research, and there's no reason to try to relearn all that chess lore from scratch

yourself by losing a lot of games for the next fifty years.

This is a common affliction in software engineering, too, sometimes referred to as the not-invented-here syndrome, but it's really worse - it's just making stuff up and thinking it works. Like the can't-lose chess opening, I've many a cobbled-up "algorithm" that just works for one or a few contrived cases and will break under real usage. There are decades worth of algorithms created in academia and industry, and research on complexity that can let you know what's practically solvable and what's not (calculating all the possible outcomes in a fifty-move chess game is an example of not). So there's no excuse not to start with prior work, and you won't embarrass yourself later.

## **7. Keep Your Ego in Check**

I've been on a couple of high school chess teams where the person who thought he was best assigned himself the lead position, and then got skunked in tournament play. Once this occurred even after a holding an internal competition to sort out the rankings - I won, but the seniors couldn't believe that a sophomore was their best player, so they discarded the results.

This happens a lot in software, too. Everyone wants to be the CTO, software architect, or at the very least, lead programmer. But, obviously, not everyone is qualified.

## **8. Know Your Pieces**

Chess players should recognize a relative valuation of their pieces: pawns the least valued, followed by knights and bishops, then rooks, the queen, and of course the king is critical. But more advanced players take in account the special capabilities and limitations of each piece. Pawns can only threaten two squares but can be extremely valuable when in position to queen. Bishops have range but are limited to either black or white squares - if you just have a bishop in your endgame, there's not much you can do. Knights can leap over obstructions but have limited mobility on the edges of the board ("knights on the rim are grim").

I've seen plenty of projects where people were placed in positions where they were ineffective, including myself.

- In my first job out of college, I was sent to another group to obtain their assistance on our project - I went back and forth several times until my manager got impatient and got what he wanted from them himself. Even if I'd been more pushy, as an entry-level hire without any authority or experience (or even expertise), I don't think there's any way I could have got the job done. And even a veteran engineer isn't all-capable like the queen. I've been on more than one project that was stalled because a senior engineer was placed in charge but didn't have the architecture or organizational skills to keep make progress.

## **9. The Sacrifice**

I hesitate to bring up the chess sacrifice, but there's a pretty obvious analogy - burnout! In chess, overeagerness to sacrifice leaves you undermanned and losing.

-