# The Cabal: Valve's Design Process for Creating Half-Life

Ken Birdwell

## Context

*Up until* ***Half-Life,*** *every major project I had ever worked on had been strictly driven by engineering constraints; I had never worked on anything whose primary constraint was that "it had to be fun." Integrating a large team of artists and game play designers into the development process was also baffling: virtually none of them had any experience working in a large complex project, and none of the senior technical staff had any experience working with non-engineers. Attempts at imposing formal methods proved pointless, and leaving each area to its own devices proved disastrous. It wasn't until we started the Cabal process that we had any hope of even completing the game, much less succeeding past all our expectations. This article originally appeared in the December 1999 issue of* ***Game Developer*** *magazine.*



Ken Birdwell is one of the founding members of Valve Inc., and contributed to Valup's *Half-Life* and *Half-Life 2* PC game titles. Previous work includes designing broadcast satellite networking at Microsoft, and automating prosthetic design tools with Nike.

While *Half-Life* has seen resounding critical and financial success (winning over 50 Game of the Year awards and selling more than a million copies worldwide), few people realize that it didn't start out a winner—in fact, Valve's first attempt at the game had to be scrapped. It was mediocre at best, and suffered from the typical problems that plague far too many games. This article is about the teamwork—or "Cabal process"—that turned our initial, less than impressive version of *Half-Life* into a groundbreaking success.

### Paving the Way with Good Intentions

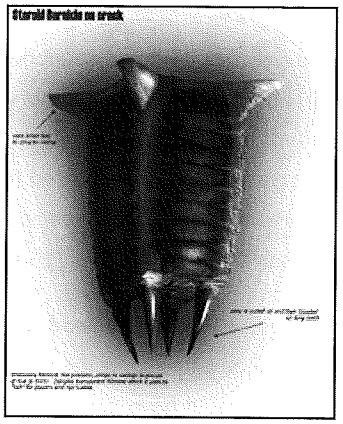
Our initial target release date was November 1997—a year before the game actually shipped. This date would have given Valve a year to develop what was in essence a fancy *Quake* TC (Total Conversion—all new artwork, all new levels). By late September 1997, nearing the end of our original schedule, a whole lot of work had been done, but there was one major problem—the game wasn't any fun.

Yes, we had some cool monsters, but if you didn't fight them exactly the way we had planned they did really stupid things. We had some cool levels, but they didn't fit together well. We had some cool technology, but for the most part it only showed up in one or two spots. So you couldn't play the game all the way through, none of the levels tied together well, and there were serious technical problems with most of the game. There were some really wonderful individual pieces, but as a whole the game just wasn't working.

The obvious answer was to work a few more months, gloss over the worst of the problems and ship what we had. For companies who live and die at the whim of their publishers, this is usually the route taken—with predictable results. Since Valve is fairly independent, and since none of us believed that we were getting any closer to making a game we could all like, we couldn't see how a month or two would make any significant difference. At this point we had to make a very painful decision—we decided to start over and rework every stage of the game.

Fortunately, the game had some things in it we liked. We set up a small group of people to take every silly idea, every cool trick, everything interesting that existed in any kind of working state somewhere in the game and put them into a single prototype level. When the level started to get fun, they added more variations of the fun things. If an idea wasn't fun, they cut it. When they needed a software feature, they simplified it until it was something that could be written in a few days. They all worked together on this one small level for a month

Many of our scripted sequences were designed to give the player game-play clues as well as provide moments of sheer terror.



Conceptual artwork for ceiling-mounted monster that was dangerous to both the player and the player's enemies.

while the rest of us basically did nothing. When they were done, we all played it. It was great. It was *Die Hard* meets *Evil Dead*. It was the vision. It was going to be our game. It was huge and scary and going to take a lot of work, but after seeing it we weren't going to be satisfied with anything less. All that we needed to do was to create about 100 more levels that were just as fun. No problem.

### So, Tell Me about Your Childhood

The second step in the pre-Cabal process was to analyze what was fun about our prototype level. The first theory we came up with was the theory of "experiential density"—the amount of "things" that happen to and are done by the player per unit of time and area of a map. Our goal was that, once active, the player never had to wait too long before the next stimulus, be it monster, special effect, plot point, action sequence, and so on. Since we couldn't really bring all these experiences to the player (a relentless series of them would just get tedious), all content is distance based, not time based, and no activities are started outside the player's control. If the players are in the mood for more action, all they need to do is move forward and within a few seconds something will happen.

The second theory we came up with is the theory of player acknowledgment. This means that the game world must acknowledge players every time they perform an action. For example, if they shoot their gun, the world needs to acknowledge it with something more

permanent than just a sound—there should be some visual evidence that they've just fired their gun. We would have liked to put a hole through the wall, but for technical and game flow reasons we really couldn't do it. Instead we decided on "decals"—bullet nicks and explosion marks on all the surfaces, which serve as permanent records of the action. This also means that if the player pushes on something that should be pushable, the object shouldn't ignore them, it should move. If they whack on something with their crowbar that looks like it should break, it had better break. If they walk into a room with other characters, those characters should acknowledge them by at least looking at them, if not calling out their name. Our basic theory was that if the world ignores the player, the player won't care about the world.

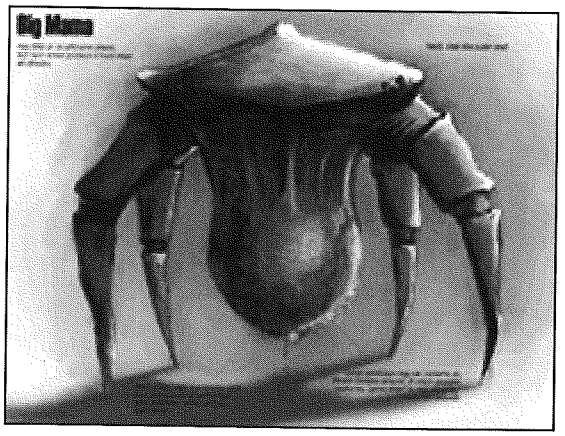
A final theory was that the players should always blame themselves for failure. If the game kills them off with no warning, then players blame the game and start to dislike it. But if the game hints that danger is imminent, shows players a way out and they die anyway, then they'll consider it a failure on their part; they've let the game down and they need to try a little harder. When they succeed, and the game rewards them with a little treat—scripted sequence, special effect, and so on—they'll feel good about themselves and about the game.

**Secret Societies**

Throughout the first eleven months of the project we searched for an official "game designer"—someone who could show up and make it all come together. We looked at hundreds of resumes and interviewed a lot of promising applicants, but no one we looked at had enough of the qualities we wanted for us to seriously consider them the overall godlike "game designer" that we were told we needed. In the end, we came to the conclusion that this ideal person didn't actually exist. Instead, we would create our own ideal by combining the strengths of a cross section of the company; putting them together in a group we called the "Cabal."

The goal of this group was to create a complete document that detailed all the levels and described major monster interactions, special effects, plot devices, and design standards. The Cabal was to work out when and how every monster, weapon, and NPC was to be introduced, what skills we expected the player to have, and how we were going to teach them those skills. As daunting as that sounds, this is exactly what we did. We consider the Cabal process to have been wildly successful, and one of the key reasons for *Half-Life's* success.

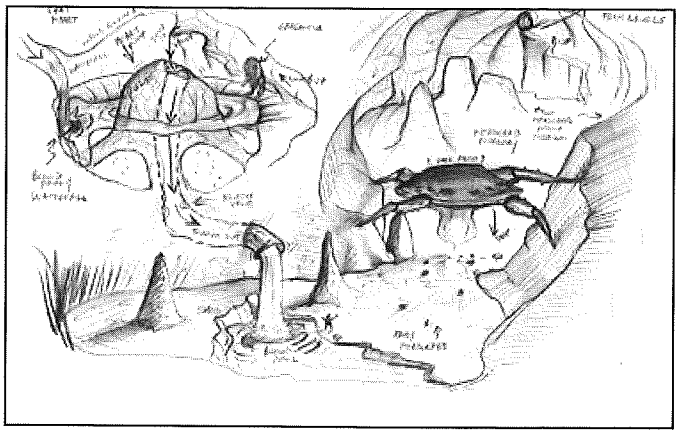
Cabal meetings were semi-structured brainstorming sessions usually dedicated to a specific area of the game. During each session, one person was assigned the job of recording and writing up the design, and another was assigned to draw pictures explaining the layout




The team explored a variety of visual metaphors that resulted in some very unique and effective opponents.

and other details. A Cabal session would typically consist of a few days coming up with a mix of high level concepts for the given area, as well as specific events that sounded fun.

Once enough ideas were generated, they would be reorganized into a rough storyline and chronology. Once this was all worked out, a description and rough sketch of the geometry would be created and labeled with all the key events and where they should take place. We knew what we wanted for some areas of the game from the very start, but other areas stayed as "outdoors" or "something with a big monster" for quite some time. Other areas were created without a specific spot in the game. These designs would sit in limbo for a few weeks until either it became clear that they weren't going to fit, or that perhaps they would make a good segue between two other areas. Other portions were created to highlight a specific technology feature, or simply to give the game a reason to include a cool piece of geometry that had been created during a pre-Cabal experiment. Oddly enough, when trying to match these artificial constants, we would often create our best work. We eventually got into the habit of placing a number of unrelated requirements into each area then doing our best to come up with a rational way to fit them together. Often, by the end of the session we would find that the initial idea wasn't nearly as interesting as all the pieces we built around it, and the structure we had designed to explain it actually worked better without that initial idea.



It's important to include information on the intended path through the level, as well as rough geometry and character placement.

During Cabal sessions, everyone contributed but we found that not everyone contributed everyday. The meetings were grueling, and we came to almost expect that about half of the group would find themselves sitting through two or three meetings with no ideas at all, and then suddenly see a direction that no one else saw and be the main contributor for the remainder of the week. Why this happened was unclear, but it became important to have at least five or six people in each meeting so that the meetings wouldn't stall out from lack of input.

The Cabal met four days a week, six hours a day for five months straight, and then on and off until the end of the project. The meetings were only six hours a day, because after six hours everyone was emotionally and physically drained. The people involved weren't really able to do any other work during that time, other than read e-mail and write up their daily notes.

The initial Cabal group consisted of three engineers, a level designer, a writer, and an animator. This represented all the major groups at Valve and all aspects of the project and was initially weighted towards people with the most product experience (though not necessarily game experience). The Cabal consisted only of people that had actual shipping components in the game; there were no dedicated designers. Every member of the Cabal was someone with the responsibility of actually doing the work that their design specified, or at least had the ability to do it if need be.

The first few months of the Cabal process were somewhat nerve wracking for those outside the process. It wasn't clear that egos could be suppressed enough to get anything done, or that a vision of the game filtered through a large number of people would be anything

other than bland. As it turned out, the opposite was true; the people involved were tired of working in isolation and were energized by the collaborative process, and the resulting designs had a consistent level of polish and depth that hadn't been seen before.

Internally, once the success of the Cabal process was obvious, mini-Cabals were formed to come up with answers to a variety of design problems. These mini-Cabals would typically include people most affected by the decision, as well as try to include people completely outside the problem being addressed in order to keep a fresh perspective on things. We also kept membership in the initial Cabal somewhat flexible and we quickly started to rotate people through the process every month or so, always including a few people from the last time, and always making sure we had a cross section of the company. This helped to prevent burn out, and ensured that everyone involved in the process had experience using the results of Cabal decisions.

The final result was a document of more than 200 pages detailing everything in the game from how high buttons should be to what time of the day it was in any given level. It included rough drawings of all the levels, as well as work items listing any new technology, sounds, or animations that those levels would require.

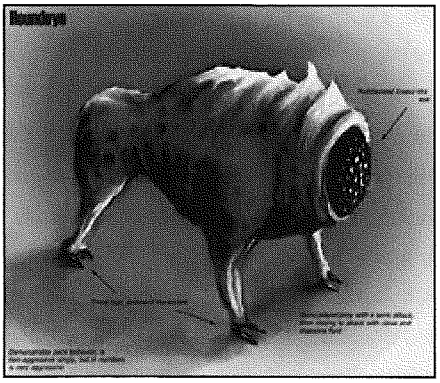
We also ended up assigning one person to follow the entire story line and to maintain the entire document. With a design as large as a 30-hour movie, we ended up creating more detail than could be dealt with on a casual or part-time basis. We found that having a professional writer on staff was key to this process. Besides being able to add personality to all our characters, his ability to keep track of thematic structures, plot twists, pacing, and consistency was invaluable.



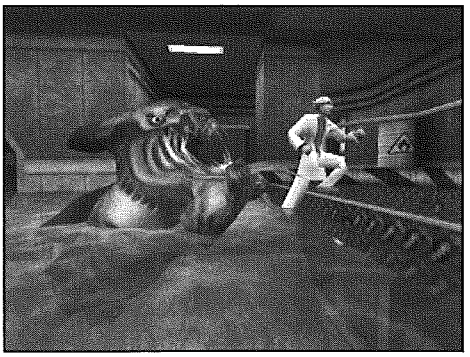
**Pearls before Swine**

By the second month of the Cabal, we (the "swine") had enough of the game design to begin development on several areas. By the third month, we had enough put together to begin play-testing.

A play-test session consists of one outside volunteer (Sierra, our publisher, pulled play-testers from local people who had sent in product registration cards for other games) playing the game for two hours. Sitting immediately behind them would be one person from the Cabal session that worked on that area of the game, as well as the level designer who was currently the "primary" on the level being tested. Occasionally, this would also include an engineer if new AI needed to be tested.



This creature was initially designed as a friendly character, but play-testing revealed players' tendencies to shoot first and ask questions later.



Letting players see other characters make mistakes that they'll need to avoid is an effective way to explain your puzzles and add tension and entertainment value.

Other than starting the game for them and resetting it if it crashed, the observers from Valve were not allowed to say anything. They had to sit there quietly taking notes, and were not allowed to give any hints or suggestions. Nothing is quite as humbling as being forced to watch in silence as some poor play-tester stumbles around your level for 20 minutes, unable to figure out the "obvious" answer that you now realize is completely arbitrary and impossible to figure out.

This was also a sure way to settle any design arguments. It became obvious that any personal opinion you had given really didn't mean anything, at least not until the next play-test session. Just because you were sure something was going to be fun didn't make it so; the play-testers could still show up and demonstrate just how wrong you really were.

A typical two-hour play-test session would result in 100 or so "action items"—things that needed to be fixed, changed, added, or deleted from the game. The first 20 or 30 play-test sessions were absolutely critical for teaching us as a company what elements were fun and what elements were not. Over the course of the project we ended up doing more than 200 play-test sessions, about half of them with repeat players. The feedback from the sessions was worked back into the Cabal process, allowing us to preemptively remove designs that didn't work well, as well as elaborate on designs that did.

Toward the middle of the project, once the major elements were in place and the game could be played most of the way through, it became mostly a matter of fine-tuning. To do this, we added basic instrumentation to the game, automatically recording the player's

position, health, weapons, time, and any major activities such as saving the game, dying, being hurt, solving a puzzle, fighting a monster, and so on. We then took the results from a number of sessions and graphed them together to find any areas where there were problems. These included areas where the player spent too long without any encounters (boring), too long with too much health (too easy), too long with too little health (too hard), all of which gave us a good idea as to where they were likely to die and which positions would be best for adding goodies.

Another thing that helped with debugging was making the "save game" format compatible between the different versions of the engine. Since we automatically saved the game at regular intervals, if the play-testers crashed the game we would usually have something not too far from where they encountered the bug. Since these files would even work if the code base they were testing was several versions old, it made normally rare and hard to duplicate bugs relatively easy to find and fix. Our save game format allowed us to add data, delete data, add and delete code (we even supported function pointers) at will, without breaking anything. This also allowed us to make some fairly major changes after we shipped the game without interfering with any of our players' hard-won saved games.

**No Good Deed Goes Unpunished**

Until the Cabal process got underway, technology was added to *Half-Life* freely. It was assumed that "if we build it, they will come," meaning that any new technology would just naturally find a creative use by the content creation folks. A prime example of this fallacy was our "beam" effect, basically a technique for doing highly tunable squiggly glowing lines between two points; stuff like lightning, lasers, and mysterious glowing beams of energy. It was added to the engine, the parameters were exposed, and an e-mail was sent out explaining it. The result was ...nothing. After two months only one level designer had put it in a map. Engineering was baffled.

During the Cabal process, we realized that although the level designers knew of the feature, they really had no clear idea of what it was for. The parameters were all very cryptic, and the wrong combinations would cause the beams to have very ugly-looking effects. There were no decent textures to apply to them, and setting them up was a bit of a mystery. It became very clear the technology itself was only a small part of the work, and integration, training, and follow-through were absolutely necessary to make the technology useful to the game. Writing the code was typically less than half the problem.

**Square Pegs**

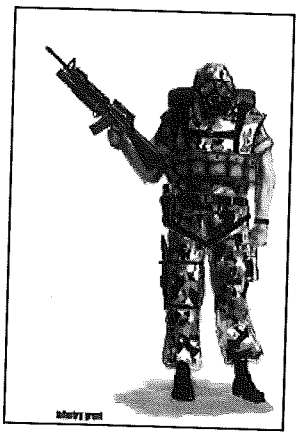
Practically speaking, not everyone is suited for the kind of group design activity we performed in the Cabal, at least not initially. People with strong personalities, people with poor verbal skills, or people who just don't like creating in a group setting shouldn't be forced into it. We weighted our groups heavily toward people with a lot of group design experience, well ahead of game design experience. Even so, in the end almost everyone was in a Cabal of one sort or another, and as we got more comfortable with this process and started getting really good results it was easier to integrate the more reluctant members. For current projects, such as *Team Fortress 2*, the Cabal groups are made up of twelve or more people, and rarely fewer than eight. The meetings ended up being shorter, and they also ended up spreading ideas around a lot quicker, but I'm not sure I'd recommend that size of group initially.

Just about everything in *Half-Life* was designed by a Cabal. This at first seemed to add a bit of overhead to everything, but it had the important characteristic of getting everyone involved in the creation process that was personally invested in the design. Once everyone becomes invested in the design as a whole, it stops being separate pieces owned by a single person and instead the entire game design becomes "ours."

This "ours" idea extended to all levels. Almost every level in the game ended up being edited by at least three different level designers at some point in its development and some levels were touched by everyone. Though all the level designers were good at almost everything, each found they enjoyed some aspect of level design more than other aspects. One would do the geometry, one would do monster and AI placement, our texture artist would step in and do a texturing pass, and then one would finish up with a lighting pass, often switching roles when needed due to scheduling conflicts. This became critical toward the end of the project when people finished at different times. If a play-test session revealed something that needed to be changed, any available level designer could make the changes without the game getting bottlenecked by needing any specific individual.

This idea also extended to all code, textures, models, animations, sounds, and so on. All were under source control and any individual was able to synch up to the sources and make whatever changes were necessary. With a little bit of self-control, this isn't as random as it sounds. It had the added benefit in that it was fairly easy to get a daily record of exactly what was changed and by whom. We would then feed this information back into the play-test cycles, only testing what had changed, as well as helping project scheduling by being able to

By placing traditional combat action in more challenging environments we were able to intensify the feeling of tension and suspense.



Placing the player in a soldier-vs.-alien conflict helped reinforce the illusion of an active environment, and let Valve show off its combat AI with minimal risk to the player.

monitor the changes and get a pretty good estimate of the stability and completeness of any one component. This also allowed us to systematically add features throughout the process with minimal impact. Once the technical portion was completed, the engineer assigned to the feature was able to synch to all the source artwork and rebuild any and all files (models, textures, levels, and so on) affected by the change.

### The Workers Control the Means of Production

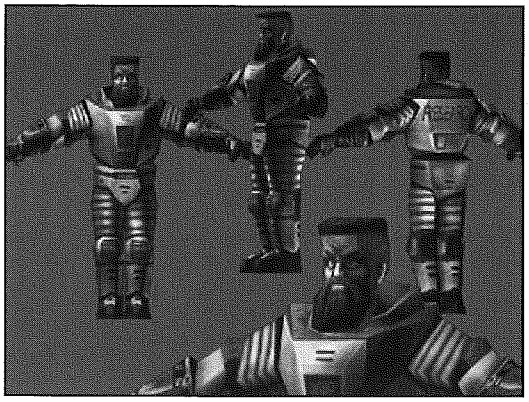
Even with all emphasis on group activity, most of the major features of *Half-Life* still only happened through individual initiative. Everyone had different ideas as to what exactly the game should look like, or at least what features we just had to do. The Cabal process gave these ideas a place to be heard, and since it was accepted that design ideas can come from anyone, it gave people as much authority as they wanted to take. If the idea required someone other than the inventor to actually do the work, or if the idea had impact on other areas of the game, they would need to start a Cabal and try to convince the other key people involved that their idea was worth the effort. At the start of the project, this was pretty easy as most everyone wildly underestimated the total amount of work that needed to be done, but toward the middle and end of the project the more disruptive decisions tended to get harder and harder to push through. It also helped filter out all design changes except for the ones with the most player impact for the least development work.

Through a constant cycle of play-testing, feedback, review, and editing, the Cabal process was also key in removing portions of the game that didn't meet the quality standards we wanted, regardless of the level of emotional attachment the specific creator may have had to the work. This was one of the more initially contentious aspects of the Cabal process, but perhaps one of the more important. By its very nature, the Cabal process avoided most of the personal conflicts inherent in other more hierarchical organizations. Since problems were identified in a relatively objective manner of play-testing, and since their solutions were arrived at by consensus or at least by an individual peer, then an authority that everyone could rebel against just didn't exist.

On a day-to-day basis, the level of detail supplied in even a 200-page design document is vague at best. It doesn't answer the 1,001 specific details that each area requires, or the countless creative details that are part of everyday development. Any design document is really nothing more than a framework to work from and something to improve the likelihood that work from multiple people will fit together in a seamless fashion. It's the Cabal process that helped spread around all the big picture ideas that didn't make it into any document—things that are critical to the feel of the game, but too nebulous to put into words. It also helps maximize individual strengths and minimize individual weaknesses and sets up a framework that allows individuals to influence as much of the game as possible. In *Half-Life*, it was the rare area of the game that didn't include the direct work of more than ten different people, usually all within the same frame.

In order for highly hierarchical organizations to be effective, they require one person who understands everyone else's work at least as well as the individuals doing the work, and other people who are willing to be subordinates yet are still good enough to actually implement the design. Given the complexity of most top game titles, this just isn't practical—if you were good enough to do the job, why would you want to be a flunky? On the other hand, completely unstructured organizations suffer from lack of information and control—if everyone just does their own thing, the odds that it'll all fit together in the end are somewhere around zero.

At Valve, we're very happy with the results of our Cabal process. Of course, we still suffer from being overly ambitious and having, at times, wildly unrealistic expectations, but these eventually get straightened out and the Cabal process is very good about coming up with the optimal compromise. Given how badly we failed initially, and how much the final game exceeded our individual expectations, even our most initially reluctant person is now a staunch supporter of the process.



The first incarnation of the game's main character, now known affectionately as "Ivan the Space Biker."

**Tips for a Successful Cabal**

- Include an expert from every functional area (programming, art, and so on). Arguing over an issue that no one at the meeting actually understands is a sure way to waste everyone's time.
- Write down everything. Brainstorming is fine during the meetings, but unless it's all written down, your best ideas will be forgotten within days. The goal is to end up with a document that captures as much as is reasonable about your game, and more importantly answers questions about what people need to work on.
- Not all ideas are good. These include yours. If you have a "great idea" that everyone thinks is stupid, don't push it. The others will also have stupid ideas. If you're pushy about yours, they'll be pushy about theirs and you're just going to get into an im passe. If the idea is really good, maybe it's just in the wrong place. Bring it up later. You're going to be designing about 30 hours of game play; if you really want it in it'll probably fit somewhere else. Maybe they'll like it next month.
- Only plan for technical things that either already work, or that you're sure will work within a reasonable time before play testing. Don't count on anything that won't be ready until just before you ship. Yes, it's fun to dream about cool technology, but there's no point in designing the game around elements that may never be finished, or not polished enough to ship. If it's not going to happen, get rid of it, the earlier the better.

- Avoid all one-shot technical elements. Anything that requires engineering work must be used in more than one spot in the game. Engineers are really slow. It takes them months to get anything done. If what they do is only used once, it's a waste of a limited resource. Their main goal should always be to create tools and features that can be used everywhere. If they can spend a month and make everyone more productive, then it's a win. If they spend a week for ten seconds of game play, it's a waste.