A STUDY IN CORPORATE CULTURES

DIGITAL EQUIPMENT CORPORATION

THE REALITY:

HEROSPEAK

A summary of what some Digital heroes have to say
about the culture.

The subject of this paper is Differentiation.

By
Reesa E. Abrams
Stephen Heiser
September 1985
Revised February 1988

Table of Contents

INTRODUCTION

The purpose of this series of papers is to show what the Digital
Culture is, has been, and is becoming in the world of the
employees who have been successful. This paper portrays six
successful engineers who emerged in the first twenty five years.
These six are preceded by myths about their accomplishments.
These are not the only six heroes of Engineering.

Heroes are very important to a culture. They provide important
information about what behavior is valued in the culture. Heroes
provide the models used by younger employees in deciding career
moves. Heroes show what is possible. Heroes become larger than
life. Every characteristic is something to be examined and
followed, especially if it gives validation to who you are and
provides you with the direction you are seeking.

There are a number of ways to study the heroes of a culture. The
way chosen was to give information about some working Engineers
in their own words. This gives the perspective of a successful
person. Rather than showing each person individually, we created
a composite of their perspectives. This hints at an accepted
Digital perspective. What is most interesting is which
perspectives were similar and which were not.

Some people may take exception to my use of the word "hero".
There is a maturity level in us all that is reached in our adult
life when we finally realize that we are the heroes of the next
generation. Our behavior is the model that will be followed.
Thus, each of us is responsible to those around us in some way,
not just for the Digital of today, but also for the Digital of
tomorrow. This is the way organizations evolve. Additionally,
in the life of an engineer, there is a difference between being
considered successful by our peers and finally considering
ourselves a success.   What I noticed about the six people I
interviewed was that they had achieved the success as well as the
maturity level. This is what a hero is all about. Furthermore,
the six heroes in this paper have continued, long-term technical
success as well as the respect of the people across all levels of
the corporation.

I had to decide who to study. This study is being funded by Bill
Johnson, so I asked him to pick six people in Engineering for the
study.

What is interesting is what they had in common. They were each clear about their technical skills and accomplishments. They each were quick to tell me how important teams were to building successful products. Additionally, each knew clearly their own strengths and weaknesses. They each told me that Digital is a production-oriented company. You must produce and keep producing to be continually successful. Each told me about the value of a mentor or some management person who kept the path clear for them to keep producing. I also heard from them how important it is to them personally that Digital is an engineering-driven company. Each of them in his own way put me through my paces to make sure I was safe to talk with.

What is also interesting about these six people is their differences. Their styles, for example, cover a broad range of characteristics. Some work lots of hours and weekends, others a regular week. Some like the intensity of New England and others want to be left alone to produce. Some think process is important, others think that getting the work out is more important than rules. They disagreed on what quality is. Some believe that it is customer-driven. Others feel that a quality product is more esoteric and that you know it when you see it. Some are arrogant, others embarrassed by all this attention. Some are affiliative, some are not. Some are introverted, some more extroverted. What stood out is that each individual has figured out what works personally. This reinforces my personal theory that one characteristic about Digital is that each person is valued as an individual. Dealing with each person is an experience in culture shock.

After I had spoken with each of the six heroes, I had a conversation with Bill Johnson to summarize his philosophy about heroes and why he had sent me after these six. The text of that interview follows:

1. Why did you pick the six people you picked?

   "Largely, because believe that within Engineering they were viewed as people who had made significant contributions over time, that they had involvement in either successful projects or products continuously, or had brought some new method or technology in to the company."

2. These are clearly the heroes of the old Digital. Where is Digital going?

INTRODUCTION 2

"First of all, they are heroes of the old Digital up to 1978. Before 1978, we had this strategy that said, there are so many markets we can go into. We are going to have so many market areas for us to go after, it is important for us to differentiate what we're doing internally and externally. Therefore, having a clear, viable objective different from anybody else at Digital was important. Differentiation was the real key to becoming a hero in the past.

"The key to becoming a hero in the future, since 1978, has been integration which means trying to make things look the same, just spaced differently."

3. "How do you get heroes in place?

"You get heroes by getting management to say and value what they do. I suggest to you that the reason why there aren't any new heroes is because there really aren't any senior managers who value that integration. We talk about one product, one company, one message, one strategy. What we really need is one really good product, and one really good company that carries with it its own message.

"Heroes can exist at all levels. I think there are management heroes that exist.

"What is interesting to me is the consistency of the messages delivered by six heroes."

Reesa Abrams
September, 1985

I. DIGITAL

WHAT IS GOOD ABOUT DIGITAL

I tend to value Digital because of the competition or the
tendency toward anarchy or the lack of central structure, and I
regard that as a valuable trait.

One of the most positive things to me has been the sense of
working with peers.

I think we're solemnly committed to building quality products,
and we have, if not a precise, at least a strong definition of
what quality means, and there's a strong desire throughout the
company to build high-quality products. We have aggressive goals
about what we're trying to achieve.

I feel a substantial sense of ownership for some of our products
and for things that have been accomplished, and I think that's
true of many of my professional peers around here. After you've
done something that you think is good, the company has put it in
to production, and it's widely used and accepted, and people are
pleased with it, then you know you did the right thing. That's
tremendously reinforcing.

Another thing that's important to me about Digital is the notion
that it's an engineering company. I really do have the sense that
the reason we're strong is because of the quality of the
engineering we do, and that provides a lot of the direction for
what we're going to do.

It's got something to do with the way the company not just says
"the people" but somehow puts its mind where its mouth is. I
don't really mean money in dollars, I mean the actions that we
all subscribe to have something to do with the fact that this is
people, even though we create machines.

If you can take it in a broad continuum, it is paying attention
to people: to the employees as people first, workers second;
customers as people first, bill paying customers second. Even
some of the things that the company does in the community have to
do with people rather than the politics of the community so much.
You sort of have a feeling that when you come through that front
door there are people who work here. That's probably the most
important thing. It translates into little things like the
creative engineering types with whom I'm most familiar. I'd give
them their head even if it is a wild goose chase or an idea that
is going to wind up costing more than it's going to benefit,
because we don't presume to know what is going to happen anyway,
so we have a little latitude in what we do.

There is a reluctance to formulate rules. We try to operate on a

minimum number of rules. We all know that once you create a rule
that concerns human behavior then the next day you're going to
have to make exceptions and eventually deviate from the
established structural guidelines. I think the one thing that is
most important at Digital is that somebody can stand up with
ideas, follow through with ideas, build products, and be the
person who guides his own destiny. That's what I really like
about Digital, and that's why I'm still here.

People like to say we're better at producing products that have
higher quality than other people, and I think that's true. I
think that is the reason we've been able to introduce new
computer architectures like the VAX and the PDP-11, that were in
the forefront.

I perceive that Digital is an atmosphere that I can excel in, and
it's an atmosphere in which I can work with good people.

We tend not to follow all the rules, and we don't chastise people
for not following the rules.

When I needed the company to come through for me, they did.

DIGITAL GIVENS

There are no absolutes.

DIGITAL CHARACTERISTICS

Running a company can be hard on people at times - a sort of ever
present fear of losing.

I don't think Digital is particularly unsafe, certainly in a
macro sense. You're probably not going to get fired or anything.
It's certainly worse other places. I think it does build up a
lot of tension.

We have indicated to our product management people that we want
to go out and talk to customers. They give us wonderful things
about how they will have the time to work on this, set up groups,
go out and talk to customers, and let us know what they say.
They're sure we're too busy to want to do this. My perception is
that there are a lot of design decisions that we need to make
that could be influenced by the customer. Existing and
prospective customers for this type of product are hard to get
because I don't know who to call.

We're operating in a vacuum. I'm guilty of this when I presume
that our customers look like us. Many people have failed on that
presumption.

We only make computers, we don't use them.

NOW VS THEN

Chances were good, if you're a middle-level technical person,
that there were only a few other people who were working in your
particular area. You have the opportunity to become a project
leader, more or less immediately, and if you do well in one or
two projects, you have the opportunity to rise and be recognized.
Very quickly. That's a lot harder at Digital these days in the
sense that we're a much larger company. We have more established
technical people, and we're also doing harder, more complicated
things. There's not that opportunity to immediately do the
technical thing and bubble to the top.

At that time we didn't try to heap so much responsibility on
product or project leaders as we do now. We didn't have the
complexities of having program managers and umpteen product
managers, and we didn't have a whole bunch of products. We
didn't have the whole company trying to inject requirements into
all the plans. Most of the products in those days were directly
related to some product line. And much of the input came from
that product line or maybe a few other product lines. There's a
lot of input now versus very little before.

HOW DIGITAL COULD BE BETTER

Digital has some legitimate superstars, but I don't consider
myself a superstar in that sense. I think there are plenty of
people who make substantial contributions who aren't superstars,
but who have something really of value to be communicated and
emulated. The company would be better off if more people were
aware of that.

I've seen examples of situations where the product is perceived
to be in trouble and a lot of turnover happens.

Projects that are somebody else's idea have a much higher failure
rate than projects that are the idea of the person who is leading
the project. One of the problems that we often have is when we
haven't identified an appropriate group to undertake a task.
Instead, we have four or more groups sitting around hacking at
the task from their own perspective.

I'm not sure how good we are at identifying those kinds of
failures and bringing them to a quick merciful end. I think
they'll tend to muddle on for a while and finally the whole thing
may just kind of collapse. One of the things around here is that
you probably end up both blaming and praising the wrong people.

Following the letter of the law is not going to make a successful
software project. There are plenty of failed projects that had
nice, thick project plans and functional specifications. They
didn't look any worse than lots of other projects and yet the

thing didn't come off at all. They did not really put the process to work in an effective way. There are people who have been quite successful by breaking lots of "rules", though I don't think there are people who have been highly successful who have just totally ignored what phase reviews were about. Frustrations are usually based on something that is keeping me from doing what I believe to be the common sense thing to do.

Periodically, when we build teams to do certain things, we don't use out heads. We build teams to give value to things and to people who are proven losers.

Today, I'm very frustrated about the fact that it takes so long to get certain things done within the company. People are so preoccupied with pettiness they don't seem to want to worry about the big things anymore. Therefore, they don't want to worry about what projects are going on and so on.

I think one of the frustrating things is that I'm a senior person in the corporation and I don't even get those memos.

The management is a cast system set up around the management people at senior group and vp level. They tend to have their staffs and the engineering people seldom, if ever, hear about certain developments.

I just wish people would use their common sense and trust me to use common sense. I think sometimes when the poison pen memos are flying back and forth, it is because we don't trust each other to use our good judgment. The biggest single problem with the corporation today is that, in engineering, people don't trust each other. Engineering people don't trust the sales force to do the right thing with products and the sales force doesn't trust engineering, so we have a terrible situation.

Another thing that we've done over the years is forgetting our roots in that we have abandoned some markets that we were "king" in. An example is a lab market. We have just let MassCom take the lab market from us. Those are the people who ran the lab business from Digital so there's no reason in the world we should've forgotten about that.

I think that today's teams are too big.

I think people have lost track of how to meet schedules because we haven't trained the people running a software or hardware project how to schedule.

I think there are a lot of people who can learn about scheduling and about how to run a software project. We don't teach them in a general way, but I think they can learn. I think that the people who are working on my project right now are learning what I believe to be a fool-proof method of how to schedule a project for success.

Someone was asked what's different about the company; why aren't we seeing more ideas come to product? The answer is because you can never get somebody to decide whether that's important or not important. We need to be able to stand up and say, "That idea is lousy. I don't want you to work on that." On the other hand, we need to be able to recognize good ideas and say, "that's a very important product for this company to be building, put a team together and do it."

The reason personnel is frustrated is because they read that darn orange book. They follow the rules and don't use common sense. They're working with people, but they're not solving people problems in people ways.

The company could help engineering get its job done by setting up a workable structure around engineering to do the things like budgeting.

When we started the VAX project, the VAX VMS, it was clearly known that we were building a team to do a specific job. And there was corporate commitment to that. We don't have corporate commitments anymore. If I was trying to get a project going now, it would be a lot easier if there was a commitment by someone who just wanted to take a stand and say, "That's important. We should do that. Go do it."

I think that the mentality of the corporation is to be all entrenched and defensive right now. But we've got to get out of that, because what made us really great was not being defensive.

We're playing catch-up all the time here. We're catching up on the hardware projects we're doing: we're catching up on the software projects we're doing.

We think we can do anything, but we are terribly constrained by realities of the corporation.

I'm not sure we do anything very consciously.

Well, we have a couple of workstations. But the problem was at that time that we had to produce the absolute Cadillac workstation that would never exist anywhere and beat the competition hands down. And consequently, we didn't get anything. I think the competition is a little better at getting a product idea formulated and into a product than we are. We've got to change that.

     If only there were some kind of a marketing strategy that lasts more than one quarter. It takes two years worth of strategy to market what we are seriously going after in a particular market. Furthermore, there must be a series of coordinated factors by which we will accomplish the goal and strategy we set out for. If not, it's disorganized.

Interactions with customers are easy compared to Digital, in

getting anything done. Customers will love you to death. Digital people will shoot you to death if you have an original idea. I'm not kidding, this company really was not invented here. It's riddled with feifdom, it's riddled with people posturing, trying to make heroes of themselves at the expense of us. We don't applaud each other's ideas at all. We attack them until, well I guess, the person is either devastated or can take anything.

If anything, we have too many good ideas, and we don't have effective ways of concentrating on choosing some of them, instead of trying to do them all, thereby not doing any well enough.

Digital is really really good at building goods, but we are hopeless at using them.

CHANGES IN THE CULTURE OVER TIME

There's certainly more overt competition between projects (than in earlier times). I guess in a sense Digital has become more dangerous. I think it is adversely affecting the culture. I think it drives people towards less sharing of information, toward less willingness to take chances. Certainly less trust.

It does seem as though there is less tendency today to break up teams and form new teams, that there is more of a tendency towards empire building, maintaining groups and that sort of thing. There's good and bad to that. In some sense it's good to maintain a stable nucleus and build on experience and all that. On the other hand, there also seems to be a tendency to do that even when you don't have a really well working group to maintain.

Because Digital is more mature, we also have the wealth and the luxury of having experienced people. Nowadays, when we start a compiler project, it would be most unusual to have the team leader be someone who has not done a large, successful compiler project for Digital before, either as the project leader or certainly as the first assistant project leader. When I first came here, we were much smaller. The language group fit into about two offices, and we didn't have that luxury, so some of us just started off being, with some brief experience at Digital and then there you are, you're the project leader of this compiler project.

STRENGTH

One of the crucial things in the success of the VAX was that it was put together as a project team or a task team and drew from diverse groups within the company that were necessary in order to pull off the first VAX product, and the whole family. We got together a group that had focus, the authority to do what it needed to do, and had the resources. in my judgment, this is probably the best technical team, perhaps, that Digital has ever put together in the sense of the number of quality people that it

had, and was able to draw on. In fact, it had people who had
been successful in previous related endeavors, mostly the
operating system or hardware design.

One of Digital's selling successes is that we are not IBM, and
people will buy from Digital because we are not IBM. They can
see through IBM's propaganda, just as anybody else could.

DO THE RIGHT THING

One of our early catch phrases for VAX was "this time we're going
to do it right" and, in fact, we had a lot of fun with that
because at various times we'd punctuate it differently.
Sometimes it was, this time we're going to do it, with right in
big capital letters and an exclamation point. Sometimes it was,
this time we're going to do it right, period. Sometimes it was,
this time we're going to do it, right?

We're engineers. We've trained ourselves as engineers through
sound schooling. Some of us have put in a dozen years at the
company and we know how to do this job. We've learned a lot.
The ones who are successful and still here have a lot of common
sense about what's good and what's bad. Trust those people and
trust yourself to make common sense decisions. So the right
thing is to use and trust each other. You know why Digital is
losing some of its good people? Because other companies know
that Digital people who are successful are very good at what they
do. And it's very hard...we get calls from headhunters all the
time and they have very lucrative offers. What they don't offer,
usually, is something that's appreciably different. I mean, it
might be more money, but it's the same old problem.

HISTORY

If people understood in a real gut way what that process was and
how it worked, I think that can be used in a lot of places. It
doesn't guarantee that every time we'll pull off a VAX, there are
only a certain number of times when (a) you're that successful
and (b) when there's such a wonderful opportunity.

    Like it used to back in 1970 when we worked on small teams in
isolated parts of the mill, making our own decisions on a very
localized basis, ignoring the people we wanted to ignore,
shooting spears out when we needed to shoot spears out. Between
1976 or 1977 and 1981 we really lost that. Groups just grew
tremendously. All of a sudden we had huge groups doing projects.
And they didn't have any direction. They were meandering. They
were perceived to be spending a lot of time watching people do
their jobs rather than letting people do their jobs. And in some
sense I think that was a reaction to managing the tremendous
growth that occurred when VAX came out. But we really didn't do
a good job at that. We put in the structures that really didn't
work. Software Engineering is a good example. It was very hard

to get things done. It was very hard to spend time working
because you were spending half your time going to meetings.
Everybody wanted to have a task force and the fact of the matter
is that some of those task forces were important. But come out
of New England and you don't get invited to any task forces and
out production level has come up tremendously. What I know now is
that some of those task forces are just a waste of time.

In the old days, which was back ten years ago, this company was
absolutely run by engineering. And I say absolutely in the sense
that it was engineers that spawned all the ideas about the
products. Once in a while, marketing would say something about,
"Well maybe we ought to have this." But engineering would spawn
the idea and engineering would go ahead.

CULTURE CHARACTERISTIC

We tend to be very proprietary about our own products and want to
hold all the cards.

WHAT SHOULD WE STOP DOING

Internally we should stop the 'cover your ass' mentality, where
everyone is worried about their own turf and about their own
project in a very short-sighted way. And this gets back to the
idea of trust. If you're going to start a project or if you're
going to work on a project you have to depend on the other people
to do their best and succeed. So we should stop being so
entrenched.

II. PEOPLE

HOW DO YOU SUCCEED AS AN EMPLOYEE

Good people make themselves. It will become evident to everyone
that they're good without their becoming exceptionally arrogant.
If you are too arrogant, people will not go out of their way to
help you; they will probably go out of their way to sabotage you.

Figure out how to use the computer. I'm surprised at the number
of people, frequently managers, who can never find the time to
learn how to use the computer effectively. We sell the darn
things and, you know, we use them in our everyday work. You find
out that so-and-so has an account on the computer and you send
them mail and it turns out that they never read it.

You have to do a certain amount of public relations with your
manager to let him know why you're of continuing value to the
company.

An employee should never become invisible.

The product they were doing worked, sold a lot, made a profit,

and people came after them to try and get them to work on the next project. They got listened to. They proposed things, got promoted, got raises, and they got stock options. Now, it's a little harder to tell. It seems that its a longer time between engineering finishing a product or project, and when its shipped and cleared, to determine its success. There's a longer ramp rate, and somehow the company seems to have gotten more self-critical, and less satisfied with its product.

Everyone on a project is 100% responsible for the product.

Somehow to be successful they need to get a mentor/advisor or some relationship.

An employee can have trouble understanding what's important versus what can be a problem because we expect them to figure that out for themselves.

We tend to prefer self-directed people. We are not heavily into managing people or telling them how to do things. We expect them to figure it out for themselves and tell us how they're doing it.

I think I'm probably more in the "good worker" category. When we did the whole VAX thing there was a tremendous amount of risk there and we all accepted chunks of it. The schedules I think we committed to were very aggressive. The objectives we had, both in terms of a quality code we wanted to produce and the level of compatibility.

There was certainly risk in there and we could easily have blown them. We spent a lot of long hours and weekends getting the work done, and it was tremendously successful. I would say, for the company as a whole, that it was an incredibly risky project.

Success, if achieved, produces several positive things. Certainly there is the personal satisfaction of a success, and I think that's a strong motivating factor for people. I think, by and large, the groups around here feel that they participate in each other's success so that, when one new project comes out, everybody in that area feels a bit better about it and feels pride in that accomplishment. A sense of accomplishment in a sort of derivative sense. Another thing that comes out of it is opportunity. Once you've succeeded, then you have the opportunity to do something else, and people are more likely to pick you to do the next key thing that needs to be done or to listen favorably to a proposal for some new project.

I suppose the advice that I would give would be, try to find a place where he can put his skills to good work, have some clear goals about what he's going to accomplish, in terms of his project presumably (I'll presume he's got a project to work) and to set clear goals that agree with his project leader or manager, and then set about achieving them. And try to do a good job of measuring himself against the goals as he goes along. Make sure that he's staying on track. I think having a good mastery of the technical

skills that are required, realizing what your skills are around
what your efficiencies are, technically finding a place where you
can put those to use and being able to learn from others.

Written rules can be a real obstacle to progress, and yet, trying
to carry out what their goals are is essential to success.

I suppose part of maturing is realizing that there are no oracles.

I don't worry about whether they can program or not. What I worry
about is can I work with this person? Is this person a reasonable
person? And can they learn? Are they willing to learn? Do they
want to work with me because they think there's something exciting
here and they want to be able to do that? That's what I look for.
You see a gleam in people's eyes and know immediately that that's
the right person for you. I don't train people in quality, but I
try to impress upon people when they set their schedules, how much
time have you left for writing a test system? When are you going
to run a test system? And when someone says, "I've just
implemented a new run-time library feature." You point to them
and say, "there's Kim Peterson over there. You give Kim a test
that will test that." We didn't do that in VMS.

We didn't have a formal test system with VMS. We depended on
another group for the UETP. And that was unfortunate because
they became second-class citizens.

I've always had very successful challenging jobs to do and I
think that I have a tremendous amount of credibility, because
I've been successful. When I say I'm going to do something,
people say, "Oh, his track record is good. I know that he can do
that." And I think that's something I've earned. I don't think
that's a reward for success. I think the success has only been
something I've earned. I think everyone who's successful earns
it. They're not entitled to do. If you wait for success to walk
in the door, it isn't going to happen.

I think we're successful because we have set up an environment
that is conducive to doing projects.

You have to play the political game, but that doesn't mean that
you have to pay attention to all this nonsense going on.

What makes you successful at Digital is to work hard, use your
imagination, use your common sense, and do the things that you
commit to do. That's what your job is.

What do I expect from my team people? They contract with me to do
a certain job on a certain date and that's what they're judged
against. And I won't let them set an unreasonable schedule. It's
my responsibility to make sure they're not setting themselves up
for failure.

If you're in trouble you should speak up and not hide it. If you
say something about it, something may be able to be done. If

you sit in your office, nothing is going to happen.

I'd say the key to success in Digital is to set your sights on reasonable goals, achieve those goals, and to think very pragmatically about what you're trying to do. We're back to products and good design. Now, what do you do to interface well with the rest of the company? You try to use your common sense again and be selective about what you listen to and what you ignore. If you see something wrong, chuck a spear. That's another good thing about working out of New England, we get very little travel from Maynard, but boy we get a lot of attention when we throw a spear.

We have to set it up so that the new employees know where to go to get information. You have to encourage people to do that. You have to encourage the people you're hiring to review their design or review their thinking. One thing that the old employees do is, we talk all the time. "I'm working on this."

"I'm having a problem." You've got to teach new employees to do that. Because that's how they learn. That's how they don't get off in a corner.

I write software very, very quickly. I never write anything down. I do it all on the terminal, and I do it so quickly that I can do it ten times over in the same time that other people can do it from start to finish. Now, those people are sometimes just as successful as I am. Sometimes, more successful. And, sometimes I'm more successful. I get the benefit of lots of iterations over the design and they get the benefit of up-front thinking. I try to tell my people that if you are the up-front-thinking kind, you want to write it all down, work all the details out and then start implementing -- that's great. But if you are the 'lots of iterations' kind, make sure that you have the capability to do that. So there are a lot of ways of getting to the same thing. Don't model the way I do it if it is not going to be successful for you. Your job is to make your dates.

If you're experienced, you tend to propose projects that you know can live with in the reality of manufacturing and sales. Hopefully you can still build forward-looking products for the industry.

The good guys tend to collect more people about them and keep on doing things. So, you can sort of see the good guys from the bad guys if you're real perceptive about what's happening.

If you want to be successful in the company, then you've got to do your job. You probably have to do more than you job. You can't just take a passive role in things. You've got to take an active role. Which means that you've got to foster ideas, maybe new product ideas, or you've got to foster innovative implementation ideas. You've got to do something where you're not just saying, "I can do that. I'll do a good job at that. Just give me a job and I'll do it." Because I don't think you can do

that and really be successful in the company. If you really want
to be successful, you've got to do that at a higher level.

You've got to put yourself in a position where you possibly could
lose.

Alliances are ambiguous, a tub of concrete. By and large, they
tend to be opposed to organizational alliances, whereby once the
organization changes, the personalities change and the alliance
drops and has to be reestablished. You operate a lot on the
basis of an understanding provided that.. it's very hard to try
to write down in words what the understanding was, you'd kill the
understanding right there.

WHY DO YOU STAY

I have an opportunity to pursue things that I think are important
and going to be valuable for the company.

The company came through with their part of the bargain after my
investment in the company. I am now feeling that the company is
investing in me.

If I couldn't guide my own destiny and work on the things that I
think are important, that is mutually important for me and the
company, I wouldn't be here.

I've been treated well and I have every expectation that that
will continue. There seems to be ample opportunity to experiment
with things that I want to do as well as do things that I'm safe
to do.

WHAT TURNS YOU ON

I tend to get my jollies about getting a product out the door.

I am a product person.

VISION

I think that more attention needs to be paid to a corporate
strategy.

The last thing I want to see is the bureaucracy get any stronger.

The culture maintainers are responsible for what they do.

We're going to become more mature and responsible in the various
organizations. We won't have a Ting Guru, or a definitive oracle
who can tell us everything we need to do, but in fact, we will
have people within the various groups who will provide the kind
of technical leadership in each area to help us to move along,
and to build the kinds of products that need to be built. There

will be, I suppose, processes something like where we will try to pull together what the different oracles are saying to be sure that it's really coherent. That's what the Local Area Systems people are trying to do and, in fact, they are being supported by the operating system.

What we need to realize is that, in each of the areas, we need to have a vision of where we're headed and a strategy to work towards.

HEROES

Being a Digital hero is being perceived as a leader on a very successful project or product.

I've developed a reputation at this point, and I think I could find someplace interesting to work if I wanted to change.

I'm smart. I go out and ask questions and talk to the people.

I'm practically always doing something. I don't sit in my office twiddling my thumbs - I go read a book in the library if I don't have anything else to do.

I'm busy. I poke my nose into a lot of areas, and I usually have something to say about them. I'm not afraid to speak up in a meeting. I apparently have some skills at running meetings. I'm quite competent in a fairly broad range of stuff. I guess I think I know what I'm doing. I can be fairly assertive or even aggressive about getting what I want.

I think there certainly are heroes in Digital, and I think the notion of the hero is important as a model for people. I think there are lots of different kinds of heroes.

Heroes have incredible technical skills and prolific ability to apply them. A second attribute is the ability to produce products, and that's something that is recognized as outstanding in Digital. There are other heroes around who might have extremely strong technical gifts, not so much the product focus, and that doesn't say they don't produce products. What they don't have is the kind of prolific involvement with products that some others have.

The style, the process they use, their ability to work with people, to work with groups.

I think that the role of people at my level, who have worked for the company for ten or fifteen years and have a lot of experience, is to pour out all their experience and guide the people who have a lot of energy to do their job. In addition, people who are the senior technical people in the company have a responsibility to drive the company in ways that make sense.

One thing that sets me apart from other people is, maybe, that I

take too much responsibility for the people. I worry a lot about their technical work, I worry a lot about what the team is doing technically, but I also think I have a lot of responsibility to keep up my end of the bargain for them.

QUALITY

Things that meet customer expectations.

I promote quality by trying to remind people that the customer pays the bills. We should be concerned with customer satisfaction instead of saving a nickel here and there in the way we design something.

Every time our machine recovers from an error and it doesn't crash, that's a customer who's satisfied.

If you build a perfect machine that no one can build after the first one, you're building a prototype. Somebody has to build the other three million of them. Someone has to assure that every one of those three million looks like the first one and works the same way, that's manufacturing. It's a difficult problem.

I'm in the business of building something that an awful lot of people are going to be pretty well satisfied with. It doesn't have to be perfect in any of the dimensions, but it ought to be pretty good in all of them.

There's nothing you can do to put quality into something once its created. Quality comes from the team that's doing it, from their vision of what they're doing and how well they can execute what their plans are.

You depend on the customer for feedback, and you want them to tell you what they need before you give it to them.

What I don't agree with, frankly, is this whole effort to try to teach people about quality; to try to give people methodology for engineering quality. It's all bullshit as far as I'm concerned. You get quality by putting teams together that do real work. You're not going to get quality by trying to paint it on.

I think I have a sense of what quality is from a software engineering standpoint that I've acquired over the years, but it's going to be very hard for me to try to define it. All I know is that when we're about to put a product out, there's a feeling you get about whether or not it's right. And if it's not right, we're going to hold onto it until we feel right about it.

I do feel that groups that deal with customers, like CSSE, play a very key role in the customer's perception of the quality of Digital or the quality of the software. I don't think you can measure it by the SPRs, but you can certainly get a feeling about it.

You get quality by good design and good engineering. You don't get it by testing it at the end. Testing is fine for the five percent of the things that you hope the customer doesn't find.

We test a business plan by looking back at our goals and constraints and say, did we meet the constraints? Because I want to be able to hold those constraints up when someone comes to me and says, "How about doing X?" "I can't do X because I've constrained myself not to do X."

I think we have quality. Our software is probably as good if not better than anybody's. Our hardware is real good, regardless of what customers like to say. Everything we design is designed to work under worst-case conditions. In fact, if we didn't have to do that, we could build things a lot cheaper. A lot cheaper.

We don't build anything that we aren't proud of. We're not going to build anything that we think doesn't work. We're not going to ship anything we don't think ought to be shipped.

Some things about quality can be measured. Some things can't. Certainly, if it works the way it's supposed to work, we can measure, and we do with both the hardware and the software products we do here. Some of the things that are a little more esoteric can't be measured. If they can't be measured objectively, they can be measured subjectively.

I really can't say enough about how I disagree with this idea that quality is measured as the difference between what you produced and what the customer thought he needed, because that's not quality.

The people who build it are responsible for the quality.

BURNOUT

There's certainly a lot of pressures to burn you out.

I have a personal computer at home. I don't use it for work. I don't log in.

I sort of jealously guard my time off. I don't commit to overtime. When the proto gets first turned on, I'll show up for flight sessions and debug or something, but generally I tend to come in at 8:30 and leave at 5:30 or 6:00. I don't do late night sessions. I don't do weekends.

There's a certain rhythm to the project. When you're first doing the planning and getting up towards phase one, you can regulate it so that things stay pretty orderly. I mean, not that you know all the answers, but the amount of work you're doing is the amount that fits into an average week. Clearly, when you begin to get to the latter parts of the 1st three months before field tests, you just have to anticipate that it is going to be a big

crunch, and, again, try to have some personal life organized to accommodate that one way or another. Be sure you take a vacation with your wife and kids before the big push for the field test. That kind of thing. There's also the realization that, even when times are worse, when you're extremely hard and things are going very badly, that, at some point, there's going to be an end to this. There will be a slower time. The product will get released.

It may be very gratifying to have worked an extremely long week and gotten a major task accomplished. That has to be done sometimes, but one does not want to believe that that's the model for life, that you're going to do that continuously.

One of the things that I have learned is not to mistake effort for progress. People burnout because they're spending effort and making no progress. Then they fail. I think burnout and failure have a lot to do with each other.

If you lose your perspective, it's very hard to recover. What I'm talking about is brinksmanship, you're constantly walking on the edge of burnout. If you're working to your full potential, you're constantly walking on that edge where you could fall off and you stay on the edge by keeping your perspective, by making sure you're making progress when you're making an effort.

I think Digital burns people out. People get burned out because they work their ass off, and they finish something and they say, "Where's the rainbow, where's the pot of gold?" And they look around and nothing happens. Just nothing happens. And they say, "Why did I work that hard, that long?" They don't know they're doing this. They don't know that they really expect some praise or glory at the end. It just isn't there, and they say, "Geez, why should I do that again?" I think that's why people burn out, and I've seen quite a few of them do it. And I don't think burnout is necessary.

It tends to be, I think more on a personal level. Isolated individuals who take the responsibility for colleagues, friends, comrades at war.

DISCIPLINE

Creativity is important, but too much of it without any discipline is chaos.

I think there are processes, discipline processes, and if you don't have those, you will indeed fail.

HOW TO BE SUCCESSFUL

In my particular case, I recognized it when I got here. I started working in this office that the company had finally come through with for me after years of being frustrated over the fact

that I was getting screwed at every turn. It took a long time
for me to overcome that, like eight years for me to realize that
the company could come through for me. Up to that point, I
always thought of myself as kind of a peon. I think a lot of the
new engineers think of themselves that way and have to overcome
it because, if you look back on your own career, you have lots of
successes that you have to identify and to buoy up to your
personality at any particular moment.

I'll also make the judgment that, if you need more than twelve
people to do a project, the project is too big; you're biting off
too much.

    If you decide to work on a project and it's going to take longer
than two years, you're doing the wrong project, and taking too
much time to do too much.

I think we each have a responsibility that if we do see a
problem, to speak up about it.

I think that we're successful because we have set up an
environment that is conducive to doing projects. What gives us
grief is that we have found ourselves, after three years, to be a
bit out of touch with the day-to-day operations of the company
and what's going on, who the people are.

To be successful you should never assume that the people around
you have the answers, especially when it comes to the management.

Always know that no matter how long you've been here, or what
your title, or where you are in the pecking order, your ideas can
prevail, as long as you realize that it's 90% sweat, blood, and
tears.

We have an oversupply of good ideas and bright people who can do
the 90% innovation. Our greatest resource is people who can do
the 90% innovation to the technical side of their ideas. They're
awash with good ideas. Other parts of the company might be quite
different. I'm suggesting that perhaps it should be. Somehow in
this high-tech culture, I don't just mean Digital, I mean in the
media, people have an inordinate and undue respect for the bright
idea. Einstein said it in, I forget exactly what he said, but he
had the ideas for general relativity. It took nine years to get
it written down and explained to his satisfaction. Einstein was
more than just an idea man. He was able to render those ideas
tractable to other people and his business. That's important.

EMPLOYEES SHOULD NEVER DO

The one thing that I would not like to see people do is lie about
their progress. They get in trouble and they don't tell me, then
there's nothing that I can do to help them or me. If they get in
trouble and they come and tell me, then the chances are that

there is probably somebody that did a little better than we
thought they were going to do, and we could probably have them
help the person out. But if they don't come and tell us, then
that's really a problem.

Well, one thing we should never do is take too seriously the idea
of managing our culture, because that can degenerate into
propaganda and people being cast out as heretics. We should
never stop changing. There are a variety of businesses we should
never go anywhere near. But also we should not be afraid to try
others -- those that even have a tiny chance of being exciting
for future businesses.

RESPONSIBILITY

I think everybody has responsibility. I don't think any one
person has the ultimate responsibility, but, ultimately, whoever
I report to is responsible.

The groups are the ones who are really responsible for the
product's success as far as the engineering side of it. We can't
do anything about the marketing or sales side of it. We're, in
fact, very disappointed that our last product has not done much
better, because we really thought it would. We thought it was a
really good product. From the engineering side, the project was
really a good project and very successful. From the sales side,
right now, it's not as successful as we thought it would be.

III. PROCESS

WHAT IS RISK

I would say that this current project is, by Digital standards, a
very-high risk project...certainly a lot of people tell me that
it's crazy, and, therefore I infer that it must be high risk. Do
I feel it? Yeah, I guess I do. I don't think that it's stressful,
I think that's what makes the job exciting. That's why I'm here.

So I feel we need some risky projects. I guess I'd feel that I
was in over my head if I couldn't at least get some grudging
admission on the part of the skeptics that it might all work.

I think the risk is worthwhile. I'm also prepared to find, in a
year, that it isn't coming together, and maybe we ought to not do
this.

I don't see myself as primarily a risk taker. I tend to see myself
as making sure that I know that the thing is going to work.

This machine is running faster than probably anybody would have thought it would have, mostly because I said it was going to run that fast. I suppose that was a risk because something could have gone wrong and it wouldn't have worked, but somehow I didn't see it as a risk. I sat down and figured out "I think it can run this fast, and, by God, I'm going to make it run this fast." Then I didn't see it as a risk anymore.

One of the kinds of risks we'd face would be not doing enough. We can be sort of complacent and slow moving, and hang on to our current customer base, and let it grow. That's probably one of the biggest risks we face. I think there is certainly risk when people start out doing new things. For example, some of the projects where we've had a lot of trouble may have been problems with the way those projects were run, but we were trying to do some new things.

I don't know that I've taken really large risks. They don't look like that, although there's certainly risk every time one commits to doing a project.

There's only one kind of real risk and that's physical injury risk. You take a risk when you go mountain climbing. Technical risk does not exist. Technical risk can always be overcome by overwork.

I think, individually, we take risks -- small risks. I think projects are filled with a few small risks such as another group not finishing their product. But again, I have to trust them to do what they think is right.

   You're risking your livelihood and your reputation with people.

I would say that I do take risks. There are always hedge risks -- gambles. There's no reward for failure.

POLITICS

I feel I spend an inordinate amount of time on politics, much more than I wish I had to.

WHAT IS GOOD MANAGEMENT

A good administrative manager has to be someone who is in touch enough with what people are saying and doing to understand the realities of what is going on.

I think the downfall of a number of managers is convincing people above them that they have everything under control, then eventually the rude shock hits that things have fallen apart. I think the failure is to continually present the impression that things are under control while, in fact, they're quietly going to hell in a handbasket.

A good technical manager has to do a good balancing act. Give people enough freedom to create solutions to problems without imposing on them, but be firm enough not to accept solutions

which that person's experience suggests will not fly. They have
to be a good sorter and they have to be able to do reasonably
well at working resolution of contrary views among their people.

Our approach has been to say, well, let's start with the
presumption that they're competent, interested, and so forth.
Start at the technical level and bring all those people into what
we were thinking about and say, "We're only going to succeed if
you help us out at this venture". By and large that's been
pretty successful. There has always been a problem area and we
said, "You know, we're going to sort that out and not presume
that those people are just a bunch of turkeys."

Managers work administrative bureaucracy. For one thing, there's
a lot of paper that needs to get filled out. The manager's tend
to work schedules, intergroup coordination, and get involved in
process sorts of things more than a technical manager. A
technical manager must try to get the best technical product they
can and to get the earliest product they can. Somehow this has
to get balanced off.

One of the clear management tasks is to provide charters that are
clear and crisp enough, and, if executed correctly, will produce
things that fit into the overall Digital environment.

Characteristics in a manager that are important to me are skills in
managing people, processes, and resources. I think in that order
also. It's important that a manager be able to deal with people,
and there are lots of styles that work. I certainly prefer those
that deal with people on a fairly adult, straightforward, and
humanistic way that value them as individuals. Managers should try
to deal with them and their problems, rather than just manage to
get the job done. So I think people management is very important.

Processes, I think are quite important, particularly in Digital
because of the lack of structure. Often, one has areas where a
lot of the problem is structural and it's important for the
manager to be able to identify that and to try to put in place
structures that will help people get the job done. When you have
an interface across several organizations, you need some
communication channel and working process to help both transmit
information and resolve conflict. If you can set up structure
that people can understand and the mechanism for doing that, it
is not too hard to work across organizations.

For the technical leaders, I think that the primary thing is that
the person have a good command of the technical knowledge and the
ability to communicate that knowledge to the people he's working
with. Obviously, the technical leader may need some of the
management skills as well. He can't be just technically
brilliant and have no people skills.

The process I would have in mind is getting the people who are
going to contribute to a product, whether it's designing, selling
or whatever, to write down, understand, and then write down again

what their goals and strengths are. What's the product you're trying to build? How long will it take to do it? What is it that will make that product successful? If it's to be a leadership product, why is it that it's going to be overwhelmingly better than anything else? If it's a beat-the-competition product, a clear picture of the competition is needed to build something that's adequate and will compete sort of on an even basis, even though it may not dominate the competition.

I think the team leaders have to be emotionally tied to their products.

I feel that people who work on my projects are contracting with me to do a certain job by a certain date. They set the schedule and I help them. But, once they set that schedule, that's what

I'm going to judge them by. That's another thing that we have forgotten about, teams. We don't judge people by accomplishments anymore. We tend to be a little easy on personal judgments and reviews. I think we need to look at what people accomplish and set some expectations for them. Then they know what they're supposed to accomplish and to expect judgment on the accomplishments. I think BJ stresses that, but I don't think we stress it enough in smaller engineering groups.

If you use common sense and make a judgment, I'll trust that you did the right thing.

It's my responsibility to make sure they're not setting themselves up for failure.

I like to think of myself as a model, but then again, I don't chastise them or judge them if they don't think the way I do. I think that's important.

You try to let people decide for themselves that it's the right thing. I try not to tell anybody. People always feel better about something they have arrived at by themselves.

I think I'm very good at running projects. I have vision.

I believe managers should view their role in life as doing everything possible to make it easier for their people to do their job. It's not that the people are there to help the manager do his job, because they're not. If we take the manager away, nothing would happen. They would still be there, would still work, and still get things done right. If you take the people away, you leave the manager. All we have is the manager, and what can he do?

A good manager is a leader.

You've got to find a way to appeal to the emotion, the religion, the ego, or the drive and capture it without really telling them. What you tell them is that we are going this way. Then you head that way, and don't even look back to see if they are coming,

because you know they are. That's a leader. A leader can always employ a manager, but it's not clear that a manager can employ a leader.

One of the things I do, especially at one or three in the morning, is think about a person whom I would like to see accomplish something. What does that person really want and how can I give it to him. How can I combine this engineering problem I have over here with their talents and desires? They are two separate things, I'm looking for a combination that works.

I don't give very many directions. One of the things I try very hard to do is not give people the answer they can't ask for at all, even though I think I know what the answer is. Provoke them to think about it in a way different from the way they've already thought about it. Inquire as to what it is they're thinking about, and how it is they thought about it. Sometimes it can be real quick.

TEAMS

If I'm telling an employee to do something stupid, then I expect to hear about it immediately and in no uncertain terms. I want it to be real clear if I'm telling him or her something incorrect.

I think you have a better chance of getting a successful project if the team is assembled top down. If the team builds itself, I think the team has to grow or evolve or something -- not be placed by external forces. You need to start with a nucleus and grow it.

I think sometimes there is a tendency, both in Digital and elsewhere, to emphasize the hero in what was actually a team effort. The focus on the hero can be good to the extent of personifying a set of values, but if the notion is given that one person is the key to producing something that was, in fact, a large team effort, I think that's bad for the culture.

The group is a very good group in the sense that everybody on it, I think, feels affirmed by the group. They feel that they're accomplishing significant things, and there's no particular feeling that one of us must be the star and get all the credit. One of the values that I hold very high on any project is that the project reach agreement, generally by some kind of consensus, on what the group standards are going to be. Once that agreement has been reached, everyone must conform to it.

The team that I am in today is just like the team I was in when I started. It's a small team. A team of people that I hope would all say they knew exactly what the product is that they are building and know exactly what their part of that product is.

You can keep track of what everybody is doing if you have a team of twelve or less people. You can't if you have a team of hundreds.

I think the successful teams are a combination of two kinds of

people. You have some people who really understand what they're doing and are proven winners, and you have another group of people who are real hard workers.

It's important not to set up a cast system in the team. It is important that the technical writers, product managers, secretarial people, librarian, and junior people, if you will, all feel that they're peers on an equal basis. There's one project leader, there's one administrative leader. Everybody else is equal and has an equal contribution to the product. It's critically important to make sure that your technical writers don't feel that they're second-class citizens and so on down the line.

If the system doesn't work, it's broken. I get upset because the system is broken not because the person screwed up. I have a right to yell and throw things too. I've never met anybody who's done something bad intentionally.

The reason the team I was on failed was because we tried to do too much.

We try to hire people who we think will fit into to the group. We try to hire people who are aggressive, who will be able to stand up and defend their ideas. We try to hire people who are ambitious. We don't want to necessarily hire people whose goal in life is to aspire to management, because, there's no future for them here because there isn't hardly any management.

It's more like a family. No rules means that you can do anything you want to do that is socially acceptable. But your responsibility is to do your job. That's your first responsibility.

Our management structure is as flat as we can get it, and we're going to stay with that management structure until it absolutely just breaks, and it's not broken yet.

There is a cast system, but the cast system is formed from technical excellence. It's formed by experience and what you've achieved, so it's not one that's formally placed. It's just there from the achievements the people have.

CONFLICT

Person A thinks this is the way it ought to be done and person B thinks that's the way it ought to be done. My process tended to be,...you and you sit down and either tell me how it is you've worked out a solution or I will tell you a solution that will work that neither of you may like.

CUSTOMER

I don't feel as closely in touch as I would like to be.

A Digital customer is someone who considers quality to be a feature more than some of our competitors, who consider that "it does more for

you to be a feature, even though it doesn't do it hte same way every time". There is a different person who comes to Digital.

It used to be that when a machine came out, we'd go out and give marketing presentations, talk to the customers, deal with real people, and sometimes learn things from them. Over the years, we seem to have been doing less and less of that. And we're getting more of our input from various marketing groups.

I go to see customers all the time. It's our collective responsibility to make sure that we're doing the right product for the customer.

I really like to hear what customers say is good, and I also like to hear what their complaints are.

But I like to talk to customers, because I think we can solve their problems. For instance, Thursday, we had a customer who just wanted to have his hand held. He sent us a list of questions he wanted answered. All he had to do, really, is read the book, but we're going to give him half a day just to hold his hand because it's the right thing to do.

I think the people we're dealing with today are just like the people we were dealing with in 1975. I think we're still selling to customers who are exactly the same. Anyway, all this talk about expanding our market isn't true. We've just found more people who are Digital customers over the years.

Well, the engineers here have done a tour of duty and talked to customers, to observe what happens out there and also to do some teaching, basically, to the folks on the front line, what it is we've just done to them and what it is they're gonna get calls about. We also send people to Europe, Canada, Australia, and Brazil. Sometimes, if we're having trouble figuring out the customer's problem, we'll just give him a call and ask him.

We have a project manager who goes out and talks with customers.

The customer base seems to be less technical then it used to be. Therefore, we tend to design things that are simple to put together and need less fiddling with. To me, that doesn't seem like a fundamental issue -- it's still a computer and still does most of the same things but, we make some trade-offs a little bit more than we used to.

DIGITAL MECHANISMS

Digital is obviously kind of a loose and open environment. You see all kinds of mail as it gets forwarded thirdhand, or things that are argued about in Notes files, and so on. I don't think there is reason for anybody who is an individual contributor and whose interested in what's going on to feel they're totally in the dark.

HOW TO GET A PRODUCT STARTED

If you've got a good enough idea, it's got to be a good idea. If
it's in the strategy, or something that fits into the strategy,
then it's easier to sell than something that doesn't fit. But if
you came up with an idea that you could show had real potential,
as far as money or return on investment, the idea should be
explored, unless it was something we didn't want to get into.

IV. COMMENTS

COMPETITION

Actually, we have eleven competitors. One through seven is IBM.
Number eight is either Japan, all of it, or AT&T. Number nine is
the other of those two. Number ten is everybody else from Apollo
to Data General -- all the rest.

And who is number eleven? Digital

THE INDUSTRY IN GENERAL

The shakeout is always coming and it is always here.

I think it's customary at Digital to pick on marketing.

TRUST

I think that there's trust in Digital among people who have
learned that they can trust somebody else.

I think people have their personal networks of people they trust
and, otheer than that, I wouldn't trust anybody. Trust in this
sense would be something like trusting them to meet their
commitments.