

Microsoft Research Plan

Advanced Technology and Business Development

Microsoft Confidential

1. Introduction

Once upon a time, it was very clear what our product agenda was - simply take the ideas which were successful on "big computers" and move them to "little computers". A great deal of effort and cleverness went toward engineering this feat - much of it directed at problems like living with too little memory, using awkward processors, and coping with the complexity of assembly language programming. Later on, we got a bit more sophisticated and the PC industry started developing some of its own unique innovations such as a graphical user interface. At present we have just about exhausted the store house of existing technology, and the days of taking something off the shelf and adapting it to our "little computers" are over. One reason is that we have already done most of it, and of course another is that microprocessor based systems aren't necessarily "little" anymore - they rank among the most powerful general purpose computers on earth.

The onward march of hardware technology is taking personal computers to new heights of processor speed, memory and general functionality per user. CPU cycles and power, however dramatic, are only one of the issues. We are also faced with the staggering potential offered by entirely new developments in other parts of the system, such as optical storage systems, new graphics capabilities, digital data broadcasting, digital video and a host of other hardware and software innovations which will change the landscape of our industry.

The challenge, of course, is to create the software which realizes the potential that this hardware offers. We have to explore and research the new software technology necessary for the PC environment of the future, evaluate the impact on our upcoming products, and then follow up with the leadership and investment of resources necessary to turn ideas into commercial reality. Microsoft has a great deal of experience with the last two stages, but historically we have not had to spend as much effort on the first. This is going to become increasingly important to us, particularly as we seek to broaden the realm of personal computing, and make it an even more important facet of people's day to day lives.

In very general terms, we have to invest in our future by doing more work in research and technology creation. The remainder of this report is on what we specifically should do.

2. What is Research?

There are a variety of different forward looking, high technology activities which Microsoft could and should be involved in. Here is a taxonomy of the different primary functions:

- **Tracking the state of the art.** We certainly need to keep abreast of the trends in our industry, and this requires having sufficient resources to monitor up and coming areas of interest, and determining their impact on our industry and our business. Recent examples of this work include analyzing the JPEG standard, reports of technology in Japan etc.

- **Advising product groups.** Another important function is advising product groups on technology. Often this means evaluating different technology options when a sudden need arises and the product group needs more expertise to evaluate the situation. This has been a large part of what ATBD has done in the past, in areas such as device independent color and fonts.
- **Technology acquisition.** It would be foolish to believe that Microsoft could develop all of the technology which we require, or that we would even want to try. It is important to be able to evaluate and acquire technology. When this technology is directed at a specific product, then it is clear that they should have responsibility for evaluation and making the deal. Often, it makes sense to buy technology outside a specific product group. In some cases, the technology is cheaper and easier to acquire if you do it early, before a product group feels the need (which was the case when we initially contacted SGI), or when the product is fundamental enough that it crosses many products (as is the case with some deals in progress).
- **Advanced development.** This category includes projects which have a greater technology content, and risk, than normal product development. Sometimes the degree of risk and new technology is quite high, and the goal is to only to create a prototype, but in most cases advanced development seeks to produce working code (usually for productization). Advanced development is often best done in the same organization that does straight development projects.
- **New technology & business projects.** There is a class of advanced development projects which includes both technological innovation in a new area for the company as well as interaction with external companies as either partners or customers. We currently have ongoing projects in this category. These are often more like a product in structure. An example project which we may do in the future would be working with a consumer electronics company on an operating system for intelligent HDTVs. A distinguishing feature of these projects is that they are oriented toward establishing Microsoft in new strategic arena rather than just being products in the normal sense.
- **Research.** Finally, there is the task of working on unsolved problems in computer science which are critical to our strategic needs in the future. This is really applied research, because we would expect to incorporate this work into products within a 2 to 5 year time horizon.

ATBD has mainly focussed on the first five activities. Our plan for some time has been to expand the group, both to be able to handle more work in the existing areas and also to expand into new areas such as research. After looking into this in more detail, we have come to some different conclusions than in the past, and the need for a research group has become a much higher priority.

At one point we had the model that we would hire a set of "experts" in various areas - graphics, natural language, and others. The total number of such people would be fairly small - say 5 - 10 - and they would help to keep us informed on new developments, advise us in their areas of expertise etc. This really covers the first two categories above, rather than actually doing research because you need to have greater resources and a specific agenda to do research - i.e. each expert would need a group. Our experience is that is very difficult to attract top ranking technical people for these positions. The job is a combination of being an analyst, a consultant

and a reporter. The best technical people are interesting in doing something active, rather than being an "armchair" researcher. The closest analog in Microsoft's parlance would be a program manager. People with the same qualifications as an MS program manager, given appropriate access to technical experts, would be great at this, but you can't get the experts to hang around just for this. Do-ers want something to do.

Another problem with this model is that it does not give us any ability to do research, and there is an intrinsic need for certain hard problems to be solved. In the course of the next couple of years our development organization will need access (and in some cases exclusive access) to technology which does not exist today. Although we are quite capable in creating products, our developers are not equipped for this task by themselves. This isn't just for extra credit, in many cases it is needed just to fulfill the vision and commitments that we have already embarked upon. There is a lot of technology that we can and should count on getting from outside - either university research or outside companies, but there are also some areas where we need to, or have an opportunity to, create new technology ourselves.

This has lead us to put more emphasis on building a group to focus on these problems. This would include world class experts, as well as sufficient other resources so that they could actively work on applied research problems which we select as high value contributions to our strategy. They would also be available as a resource to the company for the other categories above, but this would be a sideline rather than the primary job.

2.1. Advanced Development

One of the key distinctions which has to be made is the difference between doing advanced development, product development and research. Many companies confuse these issues to their detriment. We believe that Microsoft needs to have a full spectrum of development.

The table below covers several different points on that spectrum. The three key characteristics which can be used to distinguish them are the degree of technical risk and the relation to other projects or groups. Technical risk in this context does not mean the engineering task of writing the code and fixing the bugs, but rather the question of whether we know up front how feasible the project is, and whether new techniques will have to be developed in order to achieve our goals. This sort of risk is deliberately low for most of our product development - we do not usually set out to do a product without having some confidence that it can be done. There are always surprises, but by and large the Excel 4.0 team does not doubt that they can make a great spreadsheet, or that they will have to push the frontier of knowledge in computer science forward in order to do so.

At the other extreme, research is by definition aimed at solving problems which we do *not* know how to solve. Obviously we would not start them if we didn't have some confidence that we could make progress of some sort, but you must acknowledge that there is a fair amount of uncertainty. Sometimes you don't whether you will succeed, other times you feel sure that you can do something, but you are not quite sure what it will turn out to be. The biggest uncertainty is usually time - you might feel that the problem is soluble, but it is often difficult to estimate when you will manage to solve it.

ACTIVITIES	TECHNICAL RISK	RELATED WORK	GROUP INVOLVED
Current product development	Low. Uses known ideas.	Other products in business unit.	The "status quo" product development groups.
Future versions & concepts	Medium. Tries to apply new methods.	Follow on to existing product.	Product groups or advanced development groups in same business unit
New/ advanced product exploration & prototypes	Medium. Key ingredients exist, but need integration.	May be related to existing products, or not.	Existing product groups, new product groups dedicated to the area, advanced development groups.
Advanced technology used across products.	Medium - High.	More related to the technical topic than any one product.	Advanced development groups. Must be responsive to all "customer" groups.
New technology and business development	Medium - High. Usually there is no existing model to follow.	Primarily driven by technology in its domain, and by external partners.	Advanced development groups combined with business and technical strategy.
Research	High to Very High. New approaches must be invented.	State of the art research, wherever it is found.	Research group.

In between the two extremes are a number of forms of advanced development. Those which are closely related to product groups should certainly happen in these groups. The NT project is a classic example of something that started out as an advanced development project and which matured into a product effort. Other projects, like Pen Windows may be associated with groups which are not directly building the base product. In this case, it is being driven by the Applications Division which wants to use Pen Windows for pen specific products rather than the Windows group itself. Multimedia Systems is a similar in this respect.

I believe that we need to increase the amount of advanced development work that happens in the company overall. This has direct benefits in terms of making our products more innovative and competitive. It is also a primary way that research work will find its way into products. Some work of this nature will be associated with the New Technology Projects group in ATBD, but in general most of it has to occur throughout the company. We need to think about how to stimulate this on a company wide basis. It is a very important issue, but beyond the scope of this report.

3. What's in it for the Company?

There are many examples of companies not getting very much out of their research departments. A lot of people ask, "What about Xerox PARC, or IBM's Watson Research Center, or Apple's Advanced Technology group?". There are many examples of good research which does not benefit the company involved, often to extreme proportions. Xerox is an example of a company that did the right stuff at the right time with the right vision, and still lost. They invented GUI, and yet it never did Xerox any good. IBM Research has won several Nobel Prizes for fundamental discoveries¹, but it is not at all obvious that IBM gets direct commercial benefit from research overall. The work they do is quite good - they invented high performance computer architecture (pipelining and every major idea except those that Seymour Cray had), optimizing compilers, RISC architecture, SQL and relational databases, the DES encryption scheme, arithmetic coding for data compression and other major advances. I think that the case can be made that a lot of benefit is derived from this work, and IBM does get *some* of it, but it is also quite clear that most of the benefit winds up going to others (Oracle in the SQL case, MIPS & Sun in the RISC case, the world at large for DES...). In the case of IBM and Xerox the work is good, but the connection to products is weak. There are also companies like Apple who spend a lot of money, but it is not clear to an external observer what, if anything, is going on. They have over 300 people in their advanced technology group, and have some amazing toys (their own Cray XMP for example), but nothing major has come out yet². Meanwhile, the advanced development work at Apple which has had a commercial impact, such as Hypercard, was done largely by one guy with a couple assistants - and he left the company last year. Maybe they are doing good work, but can't transition it to products, maybe they just play around - it isn't clear.

The first answer to this is that when it comes to accounting for success and failure, *research is no different than any other corporate activity* - there will always be some spectacular failures. Every aspect of business is mismanaged by somebody, and it is not at all surprising that research is among them. When people focus on the question of "why doesn't corporate research work?", and use examples like those mentioned above, they are almost always overlooking the fact that you could equally "prove" that finance, marketing, advertising etc don't work either. Look at start ups - you could point at the wreckage of a thousand valiant efforts and dismiss them too. As a class they are very risky, yet many do succeed - enough so that the PC industry is lead by companies which were start ups only 10 years ago.

The most famous examples of people not being able to transfer research to the development organization are actually not surprising when you look at it in detail. *In many cases there was no development group to speak of to give the work to!* This was certainly the case at Xerox PARC - the computer product side of Xerox was nowhere near as strong as PARC. It was not a case of having two top notch teams that couldn't agree - they basically could not get out of their own way as far as developing products was concerned. The same is true for Bell Labs - their best work was done at a point when it was *illegal* for AT&T to be in the computer business. More recently, this has changed, but AT&T is still not very competent in any sort of development or marketing of computers. If you can't do products at all, it doesn't much matter whether the inspiration was from research or not.

¹ Their most recent award was in high temperature superconductivity.

² TrueType is an exception to this, and there probably are some others, but the output to date is not commensurate with the expenditure.

Another factor which is quite telling is that most computer companies which have been able to afford to fund research in the past were *hardware* companies. If you look closely at the record, they have usually done very well at integrating hardware advances into their product line. The IBM RS/6000 uses a multiply chip with a very innovative algorithm³ which was developed at Watson Research center. It is the primary factor in the excellent SPECmark performance they quote, which is heavily floating point intensive. Nobody else has anything like it - it is a clear case of taking research to the market in a timely and effective way and getting a decisive advantage because of it. The same thing is true for every major hardware company including HP and DEC. Xerox has done many very neat things at the Webster, New York hardware lab. These have gone into products, at least in part because Xerox does have product groups in those areas. Research does work at these companies, but it is certainly more difficult for them to get mileage out of software research than hardware. My belief is that this is directly proportional to the phenomena that it is hard for them to deal with most software products at all (apart from a couple of limited categories).

One final point is that somehow people seem to feel worse about great research sitting idle than the bottom line would indicate. Xerox's failure to capitalize on PARC is certainly a shame, but from any financial or strategic standpoint it pales in comparison to the \$1 billion (in 1970 dollars) they lost in the mainframe business. Even worse was Xerox's strategic failures which cost them many billions of dollars worth of market share. The hard bottom line is clear - to kill your company, you need a bad product strategy, and to waste a lot of money you need a bad product group. The actual cost of research never amounts to much in that context. This is not an apology for doing a poor job at technology transfer, but one should keep things in perspective.

The fact is that research *does* work at a lot of companies. When research fails it almost never is because of an intrinsic problem in research itself (i.e. the inability to think of something new). Instead, the research usually falls prey to problems that can be traced to general management issues - having the right goals, transitioning technology to company benefit etc.

This discussion sets the stage for the second answer to why research doesn't always work - you need to set your goals clearly up front. A great deal of the research which does not pan out is limited by things that could of or should have been determined up front:

- What are the deliverables to the company? Sometimes the goal is to advance the frontier of knowledge, sometimes it is to let management feel like they are good corporate citizens, and sometimes it is product related. You can't do all of them well at once, and sometimes they are mutually incompatible. It is important to recognize what research is trying to deliver to the rest of the company.
- Suppose that a project succeeds - will anyone care? Many research problems are plagued by the fact that the team doing the work is focussed 100% on whether and how they can achieve their immediate technical objective, and nobody is concerned about whether this objective would be a good one to achieve.

³ It calculates $A \cdot B + C$ as one operation rather than doing add and multiply separately. It does this in 2 clock cycles, where each value is 64 bit floating point.

- What is the mechanism to transfer technology to products? At many companies the basic attitude is that they'll figure this step out once they get to it. Unfortunately, this is often too late, or the process of figuring it out is painful enough that projects die in the political fighting that ensues.

Our plan for setting up a research effort at Microsoft is to address these issues up front and build them into our system. This is not a magic formula for success - that does not exist for research any more than it does for product development. If the structure is right, the good news is that it is no more difficult, and indeed not much different from, product development and we can use our experience there to guide us.

3.1. Deliverables

The first question above is about deliverables - what kinds of things does the research group contribute and create? At the onset, we will eliminate status, image and philanthropy - those are not within the purview of this report. This leaves technology and its effect on the company. An initial question to ask is why us? What are the benefits that accrue uniquely to the people that undertake the research. There are many specifics, but the basic business benefits generically fall into three categories - you get something *early*, you get it *at all*, and you get it *period*. In more detail these are:

- **Time advantage.** A key reason to get involved in a certain class of project is that it will allow you to surprise the competition, or deny them the opportunity for a surprise. This is a weak form of access - instead of either having it or not, you have it early or late. Effectively using this time lead depends on having an efficient way to take the deliverables from research and getting them into products.
- **Access to strategic technology.** There is a common pattern which repeats over and over in our industry. Market and technology conditions evolve until suddenly a new technology is thrust into the limelight and it becomes a make or break issue. Outline fonts, RISC processors and handwriting recognition are all recent examples. In some cases there are many alternatives, but in others the only way to get access to the strategic technology is to do it yourself, because the people that developed it are content to use it as a weapon.
- **Ownership & education.** Successful research creates intellectual property which is usually owned by the creator, and it also creates experience and know how within the organization. This is the thing you "get" directly out of research, but the big question is how uniquely you get it. Discovering a fundamental truth doesn't help if all you wind up owning is the copyright on the article that tells the world about it, of if others have time to invent alternatives. Just owning something is not much of a win unless you do so uniquely at least for some period of time. Know how in an organization is unique to that organization, but of course you need to have a way of capitalizing on it.

You shouldn't start research in an area unless there is a strong chance of getting a unique edge in one of these three ways. This sounds very basic, but most research done in industrial research labs does not qualify. This is not just bad luck - you can in most cases predict this long before starting. MCC and other research consortia inherently do not offer much to their members in this regard. It is not impossible for members to get any benefit, but it is tough because the difference between being a member and a non member is not sufficiently large, members compete with each

other, the research that is done must by its nature be directed out far enough in advance of the market not to conflict with members work (which makes it hard to get a time advantage), and so forth. It is possible that a place doing really hot work could make the consortium approach work, but only if the work is compelling enough to overcome the barriers that the structure makes to the three points above.

The real question to ask ourselves is why should Microsoft do a particular research problem rather than letting others explore it? What is our value added, and why we will be able to turn the raw technology into a lasting benefit which is unique to us? There has to be a reason that we get a benefit, such as through time, strategy or ownership, which we would not or could not get if we simply let academic research or other companies pioneer the area. As an extreme case, this is why we probably won't spend a lot of time proving math theorems as an end unto themselves. No matter how fundamental the theorem, the transition between proving it and applying it is so great that a third party would be just as likely to capitalize on it as we would.

As another example, consider optical character recognition. It is clear that this will be very important going into the future, because we will need to bridge between the analog world and the new digital world which is being created. OCR is still a poor candidate for Microsoft to research⁴, because it is highly unlikely that we would get any of the advantages above:

- It has a generic interface. It is easy to treat OCR as a black box - bitmaps from a scanner go in one side and text (possibly with formatting) come out another. There is no unique advantage to incorporating this with other technology, setting an API standard, etc. At most we would own the code we wrote, some algorithms and potentially some patents (but see below).
- It is too old. There is little chance that we could get any fundamental patents in this area. We might make a technical breakthrough, but it is likely to be of an incremental nature - increasing the recognition rate from 97% to 99.9%, which is nice, but something to do in focussed product development - not research.
- It is replaceable. Because very few things get a dependency on the internal workings, somebody could come around tomorrow and replace it with a new and improved method which was utterly different. This risk is not a killer by itself, but you would have to be very certain about the time advantage you would get until this happened.

Before moving on, it should be noted that there may well be a worthwhile project in OCR lurking out there which manages to skirt these issues. The point is not condemning the field, but rather that you must confront these barriers. If somebody has an interesting new angle on the problem, then it may be well a good idea. This discussion just shows what the constraints are.

This covers the basic business issues, but leaves product and technology related benefits. These fall into several general categories. Suppose that we have done some terrific research project, and it has come to a conclusion. What are the sorts of things that the company might get out of it? Here is a list of the most important areas:

⁴ Note that it may a good idea to think early about lining up a supplier so that we can bundle it into systems when needed. The comments here are about doing it as a research project.

- Product vision and direction. Knowing what is possible, and what should be done to capitalize on it can be a key benefit. One of the primary benefits that Xerox PARC could have delivered was the vision of personal computing which developed from their work - in fact if you look at the Xerox research, a lot of it really fell into this category. The individual discoveries were good, but the overall vision was the biggest win - as the rest of the industry discovered.
- External API and format standards. Our business is driven largely on standards and one of the important contributions of new technology is creating externally visible features such as new programming models, APIs, data formats. It is possible for "black box" technology which is purely internal to be important, but it is more difficult for this to create a business opportunity for the owner or developer of the technology.
- Algorithms and know how. The most direct outcome of research is the fundamental technology which it comprises, which in computer science usually boils down to algorithms, design decisions and architectural issues.
- Patents & intellectual property. An increasingly important part of the deliverables from research are patents. Unlike the other areas in this list, patents are unique in that they do not really require any sort of urgency in getting the technology to market - as long as you file early enough and get the patent granted you have a 17 year monopoly. The translation between a patent and bottom line benefit to the company is becoming more and more direct as companies turn to this mechanism for protecting technology.
- Prototypes & code. The final deliverable in this list is the actual implementation of the research in code, which might be a prototype for a product or an actual component of a product.

This is basically in order of priority. This is not to say that the lower items like code are not important, because I would expect that each research project did in fact create code and a prototype at the very least. Nonetheless, this is not usually the major reason to do research - the code and prototype by themselves are not typically very important unless they also illustrate a new product vision, define a new programming model or draw on some of the other benefits. In certain circumstances the priorities can be utterly reversed - a patent, or the existence of a prototype to demo and show proof of concept can be crucial to business success in individual situations.

The list above may seem like an obvious enumeration of the possibilities. Like so many other "obvious" things, I believe that it is so straightforward that it is often overlooked. This serves as a kind of "check list" to evaluate a new project. Are we likely to create new algorithms? Will there be API and programming model impact?

Of course you can always be surprised in the course of investigating a topic, and one of the true joys of science is when this sort of serendipity strikes and yields unexpected benefits. This is a very powerful phenomena, and we want to encourage it by creating a collegial atmosphere for researchers to exchange ideas. I do not buy the concept that this somehow means you cannot or should not think up front about what the deliverables of a particular project are likely to be.

3.2. Will Anyone Care?

Given great deliverables, there is still a question of what the real impact of research will be. This is fundamentally a question for those *outside* the research group - the technical and strategic leadership of the company. The key thing is to be extremely focussed on getting synergy between the various research projects, and the general technical strategy of the company. The non-linear advantages that accrue when you have real synergy cannot be overstated.

The best way to implement this is by focusing the vision of the future on a couple of themes which are easy for the researchers to internalize, and identify with. You must also make sure that there are some people - essentially program managers - in the research group that can act as bridges to what the rest of the world and the rest of the company are doing. This is discussed more below.

Another important point is to focus on problems where we are likely to get a big win. This again sounds obvious, but it too is often overlooked when research is planned. There are many risks associated with planning technology to be deployed in 2 to 5 years. The dynamics of our industry is such that many predictions fall by the wayside. Nevertheless, I believe that if you concentrate on the really big wins, and analyze the risks up front it is possible to come up with a research agenda which has a high probability of success. This is really no different than the existing problem of creating a long term vision and strategy for development.

3.3. Transition to Products

Once you have created some great technology, there remains the problem of effectively transferring it to the development organization. Failure to do this effectively is a primary reason that research work is ineffective at many companies.

There is no one magic formula to mastering this process - it must be managed throughout the life cycle of the research project. Some of the important factors are:

- High level strategic support is vital. The research group and the development groups must view each other as peers, and the best way to accomplish this is via the right support for the overall strategy within the company. This boils down to ensuring that the common themes and technical vision for the company are in fact shared and common to both. This process is largely "top down" - it requires the commitment of the technical and strategic leadership.
- Selecting the right research agenda is more than half the battle. The largest single technology transfer problem is that the technology is off target and nobody wants it or needs it in their product. This is a very vital point - *no amount of technology transition "process" can help the wrong technology at the wrong time*. The criteria listed above early in the process should solve a lot of the problem.
- Proper program management keeps the agenda relevant. The process of tracking the rest of the world, and measuring research goals against our strategic needs is not just an up front thing, but has to be maintained throughout the process. This is the job of program managers in the research groups

- Communication with product groups is essential. This is another responsibility of the program manager in each research area. The development groups are their direct customers and it is important for the research group to maintain a direct channel to the program managers in the product areas. They would also be responsible for organizing retreats and brainstorming sessions which bring product people in contact with the research.

The basic model is that development groups would consider the research group to be a source of technology similar to an outside company licensing technology. They could get consulting time and so forth, but the researchers would not be expected to move directly to product development. We could have people in product groups transfer in to research and then move back with expertise, but it is not a matter of policy to move the research group wholesale into development. On an individual basis such transfers could occur of course - the issue is that it is hard to sell people on coming to the research group if it is viewed as a transitory way station which as a matter of set policy will convert people into developers as soon as their research is applicable.

We have considered (and tried to a limited extent) other methods, including the "pass through" model where people from development move into research, then back out to development to productize it. There are a number of subtle issues that have to be watched in order to make this successful. Excellent developers can make poor researchers, and visa versa. The notion of moving people with projects is nice, but it is no panacea. In the long run, a pass through structure might be a valuable thing to set up, but the primary goal at first is to build up a permanent research group which can have its own identity. Once that is sufficiently established, it will be able to absorb people in from, and out to development without changing the basic focus of the research group.

4. Structure and Organization

The basic idea is that there would be a unified research group which would report in to ATBD, and which would have sub-groups or labs which focus on particular topics. The pros and cons of this approach, and the details of how to implement it are discussed below.

4.1. Why Have a Research Group?

A question which comes up at the onset is why have a focal point for research in the company? Instead, you could distribute experts throughout product development groups which were most relevant to their work.

I do not believe that this method would be successful from a variety of standpoints:

- The best research people will not come under those terms. There are people working in universities, or at places like Bell Labs, Xerox PARC, IBM Watson Research Center, DEC's Systems Research Center who are very smart, dedicated, and interested in problems that we want to solve, but who would simply never consider going to a product group. In part this is because of the reputation that product development has in some companies (particularly the companies listed above) which colors their perspectives, and in part because there really is a difference. Either way, it is a practical barrier to hiring a lot of talented people if you insist in putting them directly in a product group. Of course by the same token there are people who want to directly be responsible for shipping products, and they would probably look askance at working in research.
- Can't create the right atmosphere. Culture and atmosphere are hard to pin down up front, but very apparent in practice. A product group which is working on a deadline and is out to nuke the competition is just a lot different than a research group - no matter how driven and focussed the researchers are.
- Synergy between research efforts is hard to obtain. This is a crucial thing to attain, especially if we want to focus our efforts on a common theme, but it is very hard to do if the research activities if they are scattered across the company. Like atmosphere, this is hard to quantify, but it is a very real effect.
- Product groups are not equipped for this. Everybody professes an interest in the future, working on new technology and so forth, but frankly speaking not everybody is good at this, or even comfortable with it once they actually get to work. Our developers are smart, that is not the question at all, but they, and their management have been selected and tuned for a different set of goals.

One could make the argument that precisely for these reasons, it makes sense to try to do it - i.e. to change the attitudes and increase the innovation in product groups. Unfortunately this is both difficult to do, and possibly undesirable. Scattering a few visionaries in the midst of non-believers who are absorbed by their product commitments is not the way to change the organization. Also, it is not clear that you want product groups to be much different than they are today - their job is to integrate and implement technology to build great products, and they are good at this. The job of research is to take unsolved problems and convert them into a tangible enough form that product groups can absorb them into products. Practically speaking, I believe that the best way to stimulate innovation in product groups by presenting them with concrete technology they could apply in an interesting way.

Of course product groups should be encouraged to do as much advanced work as they want to do. The point here is not to limit what product groups are able to do, but rather that it makes sense for the company to have a unified research group.

4.2. Proposed Structure

The basic structure is that there would be a research director in charge of managing the overall research effort. This person is similar to the chairman of a computer science department in a university (and in fact that is a potential place to look for recruits) - someone who has sufficient technical stature to be respected by researchers and technical people, but who spends most of his or her time and energy focussed on managerial duties. It is important that the research director

be able to recruit effectively for the other positions and it is vital that the director be able to instill the right team spirit and atmosphere for the team.

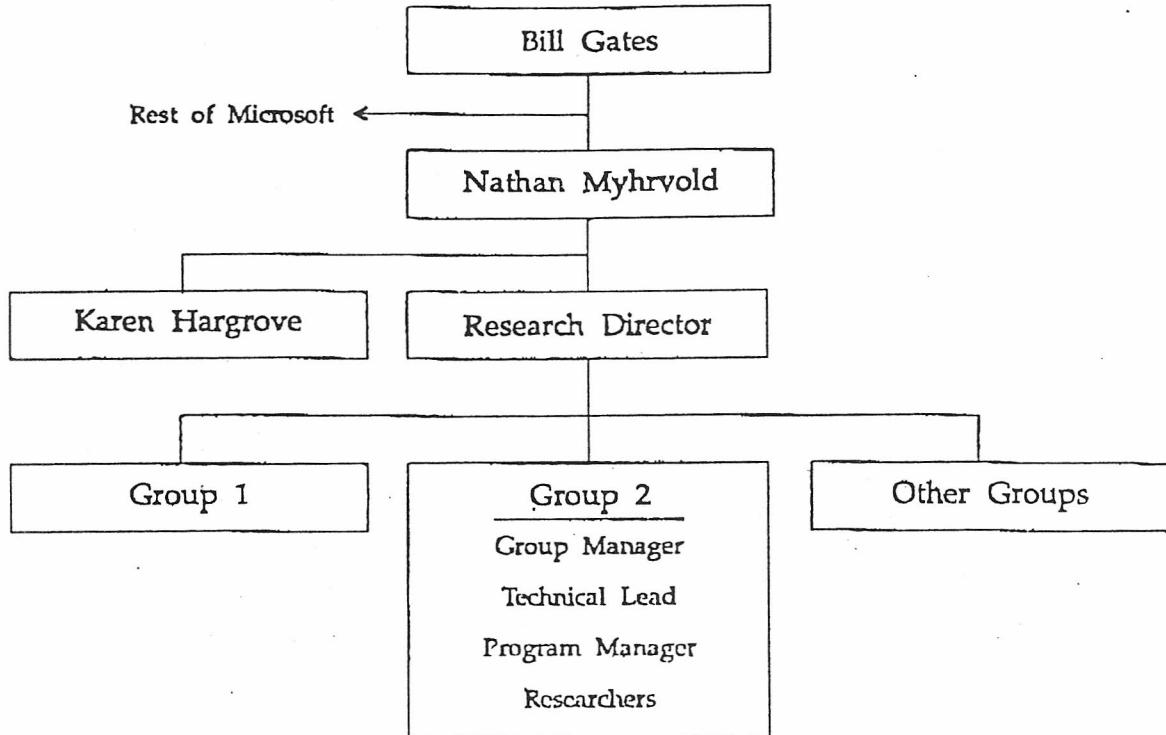
This is not to say that the research director can't think on the side, but in the first three years this is very much going to be a "start up" operation - to the same degree as a new company. A team has to be hired, a reputation has to be created from scratch, projects have to be defined and gotten underway etc. The task of leading this peculiar kind of start up is not a part time effort. A start up business usually cannot afford to have part time management once they get down to business, and neither can we. We certainly should attempt to hire a smart director who has both done research and built projects in the past, but the realistic expectation is that for the first several years at least there will be plenty to do in orchestrating the bootstrap process and that would be the directors primary responsibility.

There is a theory that says you really need two directors - a smart visionary sort of person and a "Mr Inside" who does the management and administration. The person described above for the director is Mr Inside, and this is our most important internal need. Depending on the actual people we encounter, it might make sense to have a "director" (or other nice sounding title) who did do research on the side and had more technical clout and visibility in the outside world.

Underneath the research director would be a number of research groups. Each of these would have a manager who also has research duties, perhaps as the technical leader of the group or perhaps not, depending on the group. The groups would be small enough that the managerial duty should not be too cumbersome. There would also be a technical lead and a program manager for each group. The program manager position may seem unusual from an academic standpoint, but as mentioned above, the program manager is very important for coordinating with product groups, keeping abreast of the competition and managing the research agenda. The notion of having a program manager for each research group, I believe that it is very important to achieving tangible business results, and melding the research group with the Microsoft community as a whole.

The total number of people in each group would vary from 5 to 10, depending on the area, the scope of the projects etc. Note that group does imply one project - there may be more than one project or sub project - the group would be in a given technology area such as Natural Language etc. It is possible that we would have a couple "lone wolf" researchers that would basically be working on their own, or with one additional person, depending on the situation.

The following diagram illustrates this organizational structure:



This discussion uses titles and names which are similar to those used in the rest of Microsoft. In practice we may want to use somewhat different nomenclature. As an example, we might want to call each of the research groups a "laboratory", so a group doing imaging research would be the Microsoft Imaging Lab. The precise name of the overall research group is another open issue - it is a "group", a "unit" (as in research units), a "division", a "lab"... There are lots of possible names and this needs to be thought through. At a personal level, we probably should keep the "program manager" title fixed, so that other program managers here will recognize them as one of their own, but the other titles may not be optimal.

5. Headcount & Resources

The clear message that we have gotten from people in the research community is the most significant factor in attracting people is that we show sufficient commitment to research. This comes in several forms:

- Commitment to invest sufficient resource in research to reach critical mass. It is not possible to attract people for either the research director position or even the individual researcher positions unless there is a plan to be serious. It is universally believed that you cannot be serious with a handful of people - the atmosphere and synergy that research thrives on requires a certain size.

- Commitment to fund research so that people don't have to beg or have their projects cut. This is an extremely common fear among research people. Research groups are regularly eliminated (as an example, last year Olivetti fired its whole Advanced Technology Center with just two weeks notice). Nobody wants it to happen to them, and everybody knows somebody who has had it happen to them, so the concern is very immediate.
- Commitment to have vision and be open to put research into products. Finally, there is considerable frustration at many top research places because their product groups have very little vision and are usually not interested in what the researchers have done. It sounds amazing, but just appearing to have an open mind on this area wins the hearts of many of the researchers. They want to effect products, and many places won't let them. This is a very important way for us to attract really great, and practically minded, people. PARC, Bell Labs and Watson Research center may have a lot of credibility as places to think, but Microsoft has terrific credibility for an ability to ship products which change people's lives.

These commitment concerns are right at the top of everybody's list. You cannot talk about the position without them coming up - it is more important than compensation or any other issue. The current state in the world at large is that researchers are very cautious because they have been burned, or heard of friends getting burned, so they really think in terms of up front commitment.

This is an interesting situation. If we show firm commitment on these points (mainly by looking them in the eye and saying so), then we instantly stand head and shoulders above other research establishments, and can hire the best people, or at least have a very good shot at it. If we are wishy washy or do not make a sufficient up front commitment, we are in last place because we will appear to not know what we are doing - i.e. a bunch of hackers writing for toy PCs that think they need research, but don't know what to do about it.

The first area is critical mass - having enough people working in research areas that we can get serious work done. The magic number that most people seem to quote is a minimum of 50 people. Note that we do not have to hire that many people at once - everybody understands quality control, and nobody would want to grow too fast. Nevertheless, you have to have a stated goal of reaching 50 people within two years, or you seem like a dilettante.

In this scenario, the first year the headcount for research would be 30, and the second year would be 50, with some reasonable (but smaller) growth in the third year - say 60. These would all be new heads, in addition to the present ATBD headcount.

Note that even in the third year with 60 people, the total investment in research is pitifully small for a company of our size (especially if you project what our size will be in three years). If we went by the same criteria as most Japanese companies (Sanyo, Hitachi, Ricoh, Sony etc) we would have over 300 people doing research - and that is going by our size today. I am not advocating numbers for the sake of it - but we have to remember that outsiders will judge us by these standards to some extent. They will also consider the absolute number of people it takes to get several reasonable research efforts going as a viable "critical mass".

It is not clear whether we will actually hire all of the 30 people in the first year, but it is quite possible. We have already been contacted by research groups at HP Labs, IBM, and the IRIS

group at Brown who want to leave en masse - in each of these cases we could pick up 3 - 6 people as a team in a short period of time.

I am perfectly willing to accept a contingency statement whereby the heads are not actually released until we meet milestones, such as hiring the research director etc.

6. Recruiting

Each of the research teams should be lead by world class people - there is really no excuse for settling for second best. I believe that we can get people to do the best work in the world in the areas we choose to enter.

The initial recruiting plan is as follows:

- Hire Gordon Bell as an advisor to the research group. The key initial task is to get our plans straight and then attract the right people, and Gordon would appear ideal for this. He would probably not be available to actually lead the group, but he would be ideal for finding the research director, as well as many of the technical leaders for each area.
- Consider having an advisory board. We have long discussed having a scientific advisory board, and it especially makes sense for the research group. One of the near term benefits in setting such a board up is to get the board members to assist in referring people, and to enhance our reputation. The board could include people like Gordon Bell, John Hennessy and Doug Lenat - very well known people who have had enough contact with us to be easy to recruit to the board. We would want to try to keep it small at first so that it was not a big overhead in and of itself but it could be a net positive even in the short term due to referrals.
- Pursue the director first. The highest priority is to go after the research director, so that he or she will be on board to help with the rest of the recruiting, and to manage the group as it grows.
- Be open to opportunities. Although the research director is the top priority, special opportunities may arise which warrant immediate action. The IBM natural language people are an example - we have to strike while the iron is hot in order to get them. Hiring the core team of three people can occur before the research director is hired, and in fact should lend some credibility to the effort because of their reputation.
- Go after experts once we have firmed up the mission in each category. We need to have a reasonable idea of the research agenda before going out and hiring people. Although the staffing level would be committed to 30 and then 50, we would not just open 30 reqs the first day - we want to be careful to match the research missions with the right groups, and get the right people for them.

- Target specific experts once the research area is identified. Part of the process of investigating an area for potential selection as an Microsoft research project would be to list the best people doing the work, and to directly talk to them about their work. This is the best way to get the information, and it is also a good entrée for recruiting. In the longer term we would rely on a variety of recruiting programs to attract new Ph.D. students and established researchers. An example is the "visiting scientist" position where university professors could come work at MS for a year. This is typical practice in universities and some research centers, and in Microsoft terms you can think of it as an advanced version of the summer intern program! Besides the benefit of the work that they do, this helps establish a relationship for getting their graduate students in the future.

7. Research Agenda

The primary purpose of this document is to lay out the plan for building the group rather than listing all of the research that will be done. *The discussion below is simply meant to illustrate the kinds of projects that are envisioned.*

Although there will be groups in a variety of different technology areas, it is vital to focus our efforts toward some common themes which are shared by all of the groups. This serves as a way to communicate our goals to everyone on the team, and try to channel spontaneous creativity in the right direction. Example research themes are:

- Information at your fingertips. This is to be interpreted in the broadest sense - making the personal computer into an information access and reading tool rather than just an authoring tool. The implications of this campaign go well beyond our present set of projects in this area, and will provide a lot of opportunity for research.
- The digital world. The world is going digital, and this creates enormous opportunities for integrating devices, services, and even whole industries which have been quite distinct in their analog manifestations. All current means of delivering information are suddenly going to be on common ground. The center of this cyclone is the personal computer and the software inside it. PCs are where this information will be created, and they are the vehicle though which they will be delivered. There is a great deal of research to be done in putting this together - both in how you manipulate, store and distribute the data, as well as the IAYF issues which focus on how an individual copes with it.
- Creating the digital office and home of the future. There are many interesting problems in computer science, but we want to focus on those that will become part of the mainstream of personal computing - the things which will help office workers and the ordinary "man in the street" who will increasingly rely on computing technology as a vital part of their lives.

If these sound redundant, it is by design. They are just different ways of looking at the same thing - how the personal computer will evolve between now and the mid 1990s.

In addition to the general themes, we should strive to have as much in common between the projects as we can. The strategic environment for all work will be Windows (or more precisely,

the target environment is one that supports the Win 32 API), and most of the work will probably be done on top of NT. Some of the work may wind up in future versions of Windows, some may be in applications on top of Windows, but in any event, Windows is the core. In addition to being the target, Windows would also be the working development environment. This gives us good product feedback, and it also lets us develop tools that can be shared across the group.

I would also like to see us focus on as few implementation languages as possible - ideally just one, which probably means C++ (with ordinary C as an acceptable subset). Getting the AI people to use C++ instead of Lisp may or may not be feasible (I am actually quite optimistic), but it would be nice if we could share as many tools etc. as possible. This kind of detail may sound like a nit, but it is one of many ways that you build synergy in the effort. Historically, the most successful research groups have often been very hard core about this - at Bell Labs essentially everybody uses C (and UNIX). Xerox PARC changed its mind on languages several times (SmallTalk, Mesa, Interlisp-D), but at any point in time 90% of the work was in one of them. Just about everything done at IBM Watson Research center is in PL/I and DEC SRC uses Modula 3 exclusively. Of course each of these places invented the languages they fell in love with, and I see no reason for us to do that - in the near term⁵ at least!

Another point of philosophy is that we should have *users* of the technology within the research group. That is one of the motivations behind the two research projects listed below to explore new application categories. They will not be required to use technology developed in other research groups, but there will be ample opportunity to do so, and synergy between the various groups will be encouraged. It will also be typical for research groups to make their own prototype applications or demos. We do not want to fall into the trap of designing theoretical systems years before application writers get to see them. This is precisely what we did in the case of Windows and Presentation Manager - each⁶ were designed in what amounted to an "open loop" fashion - so this is a very real concern. This phenomenon is occurring today in with IAYF - most of the thinking that is being done is in the systems components and not enough at the application level. We want to make a conscious effort to organize the research agenda so that application level thinking is fully up to speed with system issues - and ideally out in front of them.

One reason to do this is that there is an enormous amount to be learned from having the entire "food chain" interact - the application people can beat up the add-on library people who can complain to the kernel people who will rant at the people supporting exotic devices and so forth. The feedback you get is often invaluable. When we invent a new programming model, thousands of ISVs will have to live with it for years, so we want to get as much review, from all of them as possible done up front.

⁵ Using "rich source" to view the code in a very different manner might be interesting.

⁶ Especially the User component, the input model, and the window manager.

7.1. Research Activity Decisions

Themes set the tone and get people aware of the basic thrust of the research, but there is still the process of deciding what projects to do, and what areas we should investigate. The general methodology for doing this in the start up phase of the research group will be:

- **Do a lot of homework.** It is important to put a lot of effort into checking the area out up front. This includes going through the list of criteria discussed earlier in this memo, as well as reviewing the technical issues.
- **Identify existing work and experts.** Depending on the field there may be a little or a lot of relevant research being done at other places.
- **Prepare a proposal.** This will outline the general direction for research, the benefits and who we should attract.
- **Recruit team.** Note that in most cases we will expect to rely on outside experts rather than going inside the company, although that is not out of the question, especially for programmers and program managers.
- **Kick off detailed planning with a retreat.** Once we have the experts on board, the typical way to initiate the actual planning would be with a retreat that involved key technical people across Microsoft, the research team and some advisory board members.
- **Do not succumb to process or bureaucracy.** The list above may sound like there is a very formal process by which we decide to do a project. If that really occurs, then we are doomed. There has to be the right sort of dynamic involved in balancing common sense (which is mainly what this is) with formality. It is silly to open a major lab without a little homework, but if you require a ton of paper before you follow up on a spontaneous idea, then you kill creativity. This tradeoff has to be properly balanced.

In the long term one would expect that ideas and proposals come from a variety of sources - other research projects, requests from development and so forth.